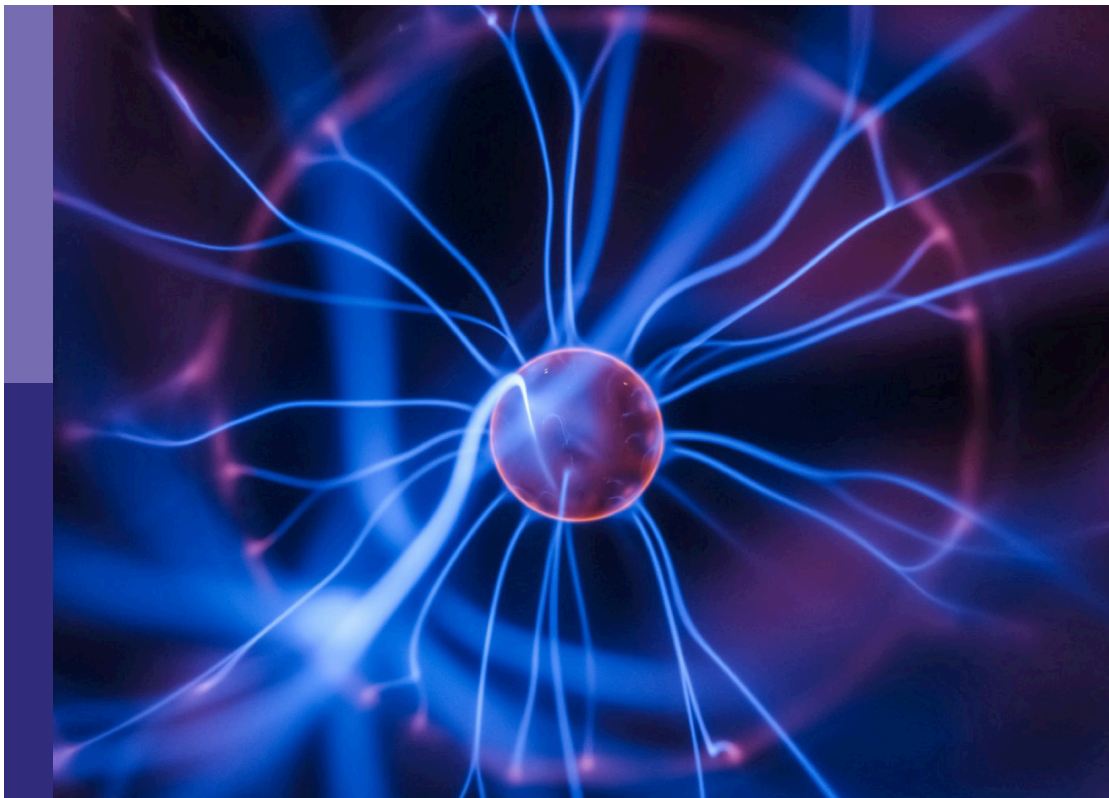
frontiers | Research Topics

# Advanced deep learning algorithms for multi-source data and imaging

**Edited by**
Jicheng Wang and Haoyu Chen

**Published in**
Frontiers in Physics
Frontiers in Medicine



ISSN 1664-8714
ISBN 978-2-8325-7115-6
DOI 10.3389/978-2-8325-7115-6

**Generative AI statement**
Any alternative text (Alt text) provided alongside figures in the articles in this ebook has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

## About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

## Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

## Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

## What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

# Advanced deep learning algorithms for multi-source data and imaging

**Topic editors**
Jicheng Wang — Jiangnan University, China
Haoyu Chen — University of Oulu, Finland

**Citation**
Wang, J., Chen, H., eds. (2025). *Advanced deep learning algorithms for multi-source data and imaging*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-7115-6

# Table of contents

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 22 April 2024
DOI 10.3389/fphy.2024.1401857



OPEN ACCESS

EDITED BY
Zhenqiu Shu,
Kunming University of Science and Technology, China

REVIEWED BY
Kun Cheng,
Beihang University, China
Jun Yu,
Zhengzhou University of Light Industry, China

*CORRESPONDENCE
Shaoping Yang,
yangsp@ynzy-tobacco.com

RECEIVED 16 March 2024
ACCEPTED 05 April 2024
PUBLISHED 22 April 2024

CITATION
Yu Z, Yang S, Li Z, Li L, Luo H and Yang F (2024), LogMS: a multi-stage log anomaly detection method based on multi-source information fusion and probability label estimation. *Front. Phys.* 12:1401857. doi: 10.3389/fphy.2024.1401857



# LogMS: a multi-stage log anomaly detection method based on multi-source information fusion and probability label estimation

Zhongjiang Yu, Shaoping Yang*, Zhongtai Li, Ligang Li, Hui Luo and Fan Yang

China Tobacco Yunnan Industrial Co., Ltd., Kunming, Yunnan, China

**Introduction:** Log anomaly detection is essential for monitoring and maintaining the normal operation of systems. With the rapid development and maturation of deep learning technologies, deep learning-based log anomaly detection has become a prominent research area. However, existing methods primarily concentrate on directly detecting log data in a single stage using specific anomaly information, such as log sequential information or log semantic information. This leads to a limited understanding of log data, resulting in low detection accuracy and poor model robustness.

**Methods:** To tackle this challenge, we propose LogMS, a multi-stage log anomaly detection method based on multi-source information fusion and probability label estimation. Before anomaly detection, the logs undergo parsing and vectorization to capture semantic information. Subsequently, we propose a multi-source information fusion-based long short-term memory (MSIF-LSTM) network for the initial stage of anomaly log detection. By fusing semantic information, sequential information, and quantitative information, MSIF-LSTM enhances the anomaly detection capability. Furthermore, we introduce a probability label estimation-based gate recurrent unit (PLE-GRU) network, which leverages easily obtainable normal log labels to construct pseudo-labeled data and train a GRU for further detection. PLE-GRU enhances the detection capability from the perspective of label information. To ensure the overall efficiency of the LogMS, the second-stage will only be activated when anomalies are not detected in the first stage.

**Results and Discussion:** Experimental results demonstrate that LogMS outperforms baseline models across various log anomaly detection datasets, exhibiting superior performance in robustness testing.

KEYWORDS

**log anomaly detection, multi-source information fusion, probability label estimation, long short-term memory, gate recurrent unit**

## 1 Introduction

Logs are vital for the upkeep of large-scale software systems as they capture crucial data produced during system operation, documenting essential details regarding server and application software activities [1–3]. With the rapid development of the information age, software systems have become increasingly intricate, resulting in a significant surge in log

data volume [4, 5]. Analyzing log data allows developers to meticulously assess system status, identify anomalies, and understand their root causes [6]. Timely detection and resolution of anomalies serve as a proactive measure to prevent system crashes and mitigate potential economic losses [7].

In the early stages of log anomaly detection, developers typically relied on manual methods such as keyword searches or simple alert rules set by log investigation tools [8, 9]. However, with the prevalence of large-scale systems today, traditional manual detection methods are no longer adequate [10]. To meet the demands of modern anomaly detection in large-scale systems, extensive research has been conducted on automatic log analysis technology utilizing deep learning [11, 12]. These technologies automate the learning of log patterns and analyze connections to identify potential anomalies effectively. Examples include LogRobust [13], DeepLog [14], and LogAnomaly [15]. Nevertheless, most existing methods focus on direct detection of log data using specific anomaly information in a single stage. This limited perspective results in lower detection accuracy and model robustness.

To overcome this limitation, we introduce LogMS, a multi-stage log anomaly detection method based on multi-source information fusion and probability label estimation. Prior to anomaly detection, logs are parsed by Drain [16] and vectorized based on TF-IDF to capture semantic information. A multi-source information fusion-based long short-term memory (MSIF-LSTM) network is proposed for the first-stage anomaly log detection, enhancing anomaly detection by fusing semantic information, sequential information, and quantitative information. Subsequently, we introduce a probability label estimation-based gate recurrent unit (PLE-GRU) network, which leverages easily obtainable normal log labels to construct pseudo-labeled data and train a GRU for further detection. PLE-GRU enhances the detection capability from the perspective of label information. To ensure the overall efficiency of the LogMS, the second-stage will only be activated when anomalies are not detected in the first stage. By modeling the correlation between log data from two stages and three perspectives, LogMS effectively mines log data to detect anomalies. The key contributions of this paper include:

1) Introducing LogMS, a multi-stage log anomaly detection method employing multi-source information fusion and probability label estimation to capture deeper relationships among log sequences, thereby enhancing anomaly detection performance.
2) Conducting systematic experiments on the HDFS [17] and BGL [18] dataset to evaluate the LogMS model. The results demonstrate the method's effectiveness in detecting various anomalous logs, showing significant improvements in accuracy and robustness compared to baseline models.

## 2 Related work

Anomaly detection techniques based on automated log analysis can be broadly classified into two categories: supervised methods and unsupervised methods.

The supervised approach [13] involves training the model with labeled training data and then applying anomaly detection on log data. However, in practical scenarios, researchers have noted that many existing log anomaly detection studies have not met expectations [19]. Most models assume a closed-world assumption, which includes the stability of log data over time and a known set of log events for training and testing [20]. Yet, due to the evolving nature of log data, unforeseen log events or sequences often arise. To tackle such log instability issues, Zhang et al. [13] introduced a novel log-based anomaly detection method named LogRobust. This method extracts semantic information from log events, transforms it into semantic vectors, and employs an attention-based Bi-LSTM model for anomaly detection. By capturing contextual information and learning diverse log event features, LogRobust effectively identifies and manages unstable log events and sequences. In fact, semantic information is a vital component in natural language understanding [21, 22], and logs can be understood as a special form of natural language. Furthermore, Lu et al. [23] pioneered a detection model based on Convolutional Neural Network (CNN) [24] in log-based anomaly detection, showcasing the potential of CNN in this domain. Their CNN-based method incorporates logkey2vec embedding, three one-dimensional convolutional layers, a dropout layer, and a max-pooling layer. Initially, log content is numerically encoded, and logkey2vec generates embeddings, which are then passed through convolutional layers with varying filters. The max-pooling layer selects the maximum feature value, and a fully connected softmax layer produces probability distribution results. In experiments on anomaly detection in Hadoop Distributed File System (HDFS) logs, the CNN-based approach outperformed Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP) methods in accuracy.

Supervised methods rely on annotated training data, which requires high data quality. However, in real-world scenarios, log data is very extensive, making data annotation impractical. Furthermore, log data also suffers from class imbalance issues, where abnormal events are usually of relatively small scale, while normal events dominate the vast majority, leading to uneven data distribution. When dealing with such imbalanced data situations, supervised learning algorithms may tend to predict normal events while ignoring abnormal events.

In contrast to supervised methods, unsupervised methods offer the advantage of not requiring annotated data [25]. This characteristic makes them well-suited for real-world environments with abundant unlabeled log data. Essentially, unsupervised methods aim to establish a baseline of normal log data by analyzing internal data correlations such as sequential relationships and quantitative associations. Any data that deviates from this established baseline is classified as anomalous. For instance, Du et al. [14] introduced the DeepLog model, which treats system logs as natural language sequences and employs Long Short-Term Memory (LSTM) networks for unsupervised log anomaly detection. The model initially learns log patterns by examining sequential relationships between log events and then uses these patterns for log prediction. This pioneering approach to anomaly detection has since been widely embraced in subsequent research. Another noteworthy model, LogAnomaly developed by Ma et al. [15], also represents log streams as natural language sequences and introduces a simple yet effective semantic information extraction method called template2vec. This method can simultaneously identify sequential and quantitative log anomalies. LogAnomaly comprises offline learning and online detection components. In the offline learning phase, templates

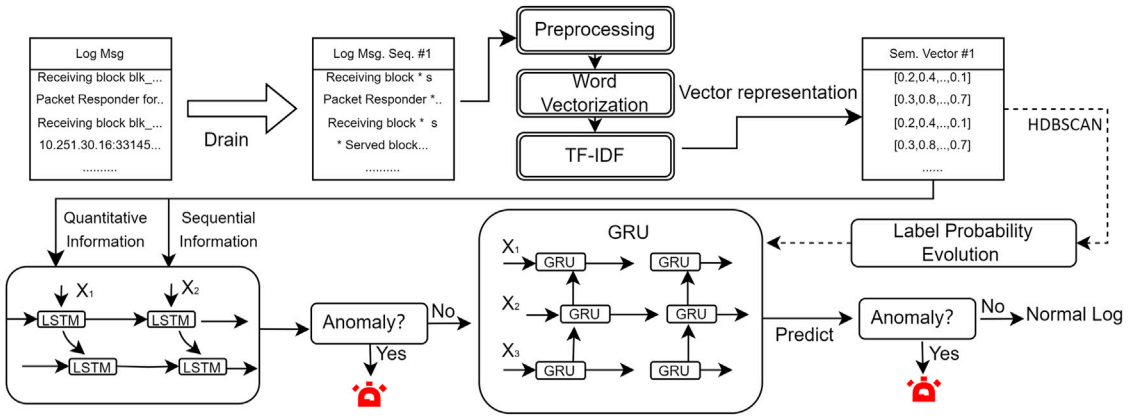



FIGURE 1
The overall architecture of LogMS.

are extracted from historical logs using FT-Tree, and the logs are matched with these templates. Subsequently, log sequences are converted into template vector sequences through template2vec. LSTM models are then used to extract sequential and quantitative features from log sequences to determine anomalies based on these features. Periodic offline learning, such as weekly updates, ensures the integration of newly emerged log templates into the updated offline models. In the online detection component of LogAnomaly, real-time logs are matched with existing templates. If a match is found, the log is converted into a template vector. Otherwise, based on template vector similarities, the real-time log's "temporary" template vector is approximated to an existing template vector. Consequently, each real-time log is associated with a template vector, and real-time logs are converted into template vector sequences. By leveraging the LSTM model trained in the offline learning phase, LogAnomaly can identify anomalous log sequences. Additionally, Farzad et al. [26] proposed a novel unsupervised log anomaly detection model that integrates Isolation Forests with two deep autoencoder networks. Autoencoders facilitate feature learning for subsequent anomaly detection, while Isolation Forests are employed for positive sample prediction.

Unsupervised methods struggle to determine the threshold range of abnormal logs, and log data typically exhibit complex data distributions, containing multiple categories and patterns, some of which may represent normal behavior while others may indicate anomalous behavior. Therefore, in unsupervised learning, without explicit labels to guide the learning process, models find it difficult to accurately discern whether logs are abnormal.

# 3 Proposed method

To address the above issues, this paper introduces a multi-stage log anomaly detection method named LogMS, which relies on multi-source information fusion and probability label estimation. The architecture of LogMS is depicted in Figure 1, which comprises the following components: Log Parsing and Semantic Vectorization, MSIF-LSTM, and PLE-GRU.

## 3.1 Log parsing and semantic vectorization

### 3.1.1 Log parsing

Raw log messages are commonly unstructured as developers have the flexibility to create free-text log messages within the source code. Hence, the initial phase in log anomaly detection involves log parsing, which aims to convert unstructured log messages into structured events. With the deepening of research on log anomaly detection, there have been many ready-to-use parsing tools that have emerged, such as Spell [27], Drain [16], Brain [28], and DivLog [29]. In this study, we have selected Drain as our log parsing tool due to its proven effectiveness and accuracy. Upon receiving a new raw log message, Drain initiates preprocessing using basic regular expressions guided by domain expertise. Subsequently, the tool searches for a log group (referred to as a leaf node of the tree) by following the specific rules embedded in the internal nodes of the tree. If an appropriate log group is identified, the incoming log message is compared with the stored log event in that group. If no suitable log group is found, a new log group is created based on the incoming log message. An illustration of log message parsing using Drain is provided in Figure 2. For instance, in the case of the initial line of the raw unstructured log message "Receiving block blk_579248908079 sc:/10.251.215.16:33145 dest:/10.251.30.6 . . . ," Drain extracts the data block name, source address, and destination address by replacing them with wildcards, resulting in the structured log event "Receiving * src: * dest:*."

### 3.1.2 Preprocessing and semantic vectorization

In this section, we will preprocess and vectorize the parsed log events following a structured workflow as depicted in Figure 3. This process encompasses preprocessing, word vectorization, and TF-IDF-based semantic vectorization.

Preprocessing: The parsed log events often contain non-character tokens (such as separators, operators, and punctuation), stop words (like "a" and "the"), and compound words (e.g., "TypeDeclaration" composed of "type" and "declaration," or "isCommitable" composed of "is" and "Commitable"). These elements can impede subsequent processes such as vectorization and anomaly detection, necessitating further preprocessing of log

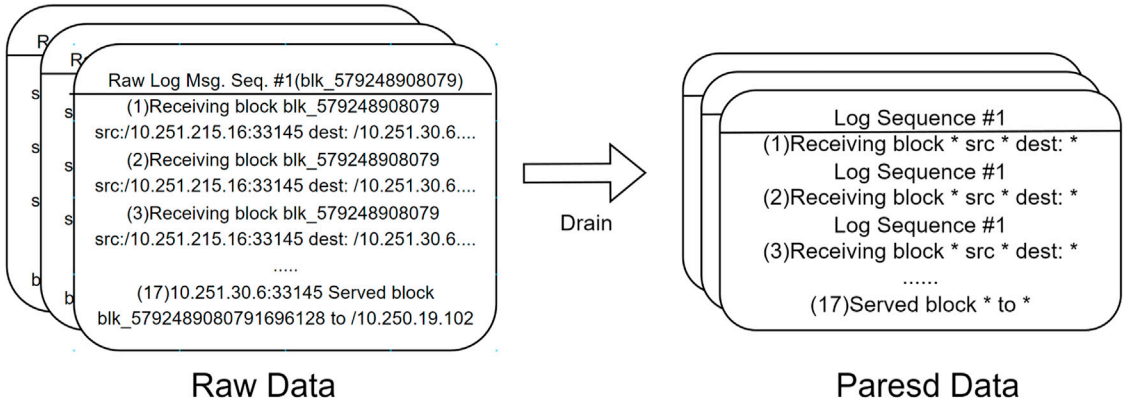



FIGURE 2
Log parsing by Drain.

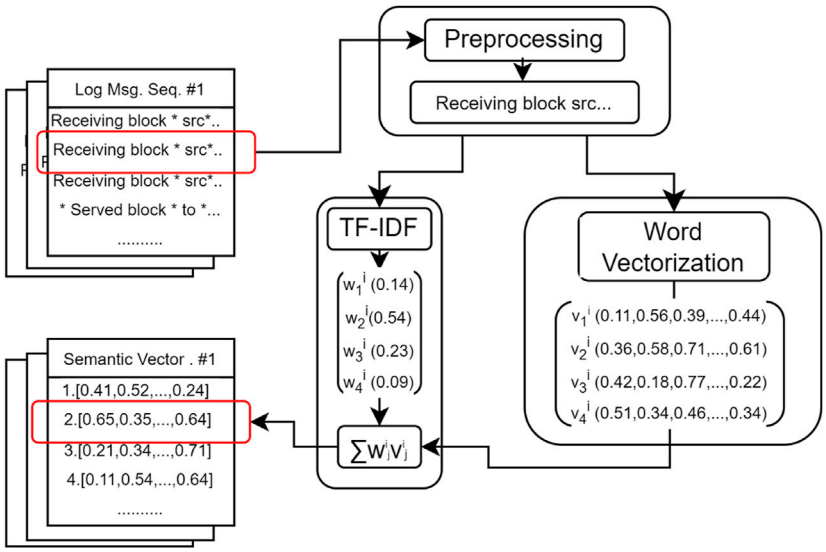



FIGURE 3
Preprocessing and semantic vectorization.

events. Specifically, all non-character tokens and stop words will be eliminated, and compound words will be segmented into individual words.

Word Vectorization: Following the preprocessing steps, each word in the log events will be vectorized with the objectives of ensuring high discriminability among different log events and identifying log events with similar semantics. To achieve this, we will utilize FastText [30] to convert words in log events into semantic vectors. FastText, pretrained on the Common Crawl corpus dataset, effectively captures intrinsic word relationships in English sentences, including semantic similarities. Implementation involves invoking the "get_word_vectors" function of FastText to acquire word vectors. The vector representation of the $j$-th word in the $i$-th log event is denoted as $v_j^i$.

TF-IDF-based Semantic Vectorization: The word vectors will be combined using TF-IDF to derive the semantic vector for each log event. In TF-IDF, the Term Frequency (TF) component gauges word importance within a sentence, promoting high discriminability. For instance, a frequently occurring word like "Block" indicates its significance. The TF calculation is defined as:

$$\mathrm{TF}\left(v_j^i\right) = \frac{\#v_j^i}{\#total^i} \tag{1}$$

Here, $\#v_j^i$ denotes the count of the $j$-th word in the $i$-th log event, while $\#total^i$ represents the total word count in the log event.

Conversely, if a term like "Receiving" is prevalent across all log events, its ubiquity may reduce event distinctiveness. To address this, the Inverse Document Frequency (IDF) in TF-IDF decreases the weight of frequently occurring terms, enhancing the weighting scheme's discriminative power. IDF calculation is as follows:

$$\mathrm{IDF}\left(v_j^i\right) = \frac{N}{N_{v_j^i}} \tag{2}$$

Here, $N$ denotes the total log event count and $N_{v_j^i}$ is the count of log events containing the word $v_j^i$. The TF-IDF weight of word $v_j^i$ is calculated by:

$$w_j^i = TF\left(v_j^i\right) \times IDF\left(v_j^i\right) \tag{3}$$

Finally, the semantic vector $v^i$ of the $i$-th log event is determined as:

$$v^i = \frac{1}{N}\sum_{j=1}^{N} w_i^j \cdot v_i^j \tag{4}$$

## 3.2 MSIF-LSTM for the first-stage dection

As mentioned above, existing methods mainly focus on directly detecting log data in a single stage using specific anomaly information, such as log sequential information or log semantic information. Therefore, we propose a multi-source information fusion-based long short-term memory (MSIF-LSTM) network for the initial stage of anomaly log detection. This method can integrate multiple information such as semantic information, sequential information, and quantitative information through multi-source information fusion. Specifically, we utilize semantic vectors representing semantic information and train the model using both sequential information and quantized information to achieve the fusion of information. We will first introduce the structure of MSIF-LSTM, and then discuss how to obtain the two information to train MSIF-LSTM. During the training process, we update the model parameters using backpropagation.

### 3.2.1 The structure and training of MSIF-LSTM

MSIF-LSTM extends the traditional LSTM architecture to handle multiple information simultaneously. The key components include the cell state ($C_t$), the hidden state ($h_t$), and multiple sets of gates - forget gates ($f_t^k$), input gates ($i_t^k$), and output gates ($o_t^k$) for each information $k$. The formulas for computing these components at time step $t$ for each information $k$ are as follows:

(1) Forget Gate:

$$f_t^k = \sigma\left(W_f^k \cdot [h_{t-1}, x_t] + b_f^k\right) \tag{5}$$

(2) Input Gate:

$$i_t^k = \sigma\left(W_i^k \cdot [h_{t-1}, x_t] + b_i^k\right) \tag{6}$$

(3) Candidate Cell State:

$$\tilde{C}_t^k = \tanh\left(W_C^k \cdot [h_{t-1}, x_t] + b_C^k\right) \tag{7}$$

(4) Update Cell State:

$$C_t^k = f_t^k * C_{t-1}^k + i_t^k * \tilde{C}_t^k \tag{8}$$

(5) Output Gate:

$$o_t^k = \sigma\left(W_o^k \cdot [h_{t-1}, x_t] + b_o^k\right) \tag{9}$$

(6) Hidden State:

$$h_t = \sum_k \left(o_t^k * \tanh\left(C_t^k\right)\right) \tag{10}$$

where $x_t$ denotes the input log event at time step $t$, $h_{t-1}$ represents the previous time step's hidden state, $W_f^k$, $W_i^k$, $W_C^k$, and $W_o^k$ stand for the weight matrices for each information gate $k$, and $b_f^k$, $b_i^k$, $b_C^k$, and $b_o^k$ are the bias vectors associated with information $k$. The symbol $\sigma$ denotes the sigmoid activation function, and $*$ signifies element-wise multiplication. The training of MSIF-LSTM involves leveraging both sequential and quantitative information. This training process not only effectively integrates multiple sources of information but also preserves the specific characteristics of each information source. This contributes to enhancing the representation capability of the model [31].

### 3.2.2 Sequential information

Logging procedures are typically executed in accordance with well-defined processes, resulting in the natural emergence of sequential patterns within normal logs. In essence, when observing a sequence of log events, it becomes possible to forecast the subsequent log event in the absence of anomalies. Therefore, we utilize the sequential information to train LSTM. The input of LSTM is a log event sequence (e.g., $\{v^{i-3}, v^{i-2}, v^{i-1}\}$), the output is the probability of the next log event.

### 3.2.3 Quantitative information

In addition to sequential information, log event sequences (i.e., sequences formed by multiple log events occurring in order) also contain quantitative information. Typically, during normal program execution, certain invariants and quantitative relationships persist within the logs, regardless of varying inputs and workloads. For example, it is an invariant fact that every opened file will eventually undergo closure at some point. Therefore, in normal scenarios, the frequency of logs indicating"open file" should be equivalent to the frequency of logs denoting "closed file." These quantitative relationships embedded within the logs serve as valuable indicators of standard program execution behavior. Deviation from these established invariants by a new log event signals an exception within the system's execution. Therefore, we utilize the quantitative information to train LSTM. First, the we need to calculate the count vector $A_k$ of $k$-th log event sequence as:

$$A_k = \left(a_k\left(v^1\right), a_k\left(v^2\right), \ldots, a_k\left(v^n\right)\right) \tag{11}$$

where $n$ denotes the total count of unique log event vectors, $a_k\left(v^i\right)$ signifies the occurrence of $v^i$ in the $k$-th log event sequence. Subsequently, $A_1$, $A_2$, …, $A_k$, … are fed into LSTM for the acquisition of quantitative insights.

## 3.3 PLE-GRU for the second-stage dection

In the first stage, the focus is on modeling some characteristics of the log data itself. The MSIF-LSTM method construct in this stage is unsupervised, lacking the utilization of label information, especially readily available normal log labels, thereby limiting the detection capability. To tackle this problem, we design a semi-supervised

learning method called PLE-GRU, which will only be activated when anomalies are not detected in the first stage, aiming to ensure the overall efficiency of the LogMS algorithm. PLE-GRU consists of three parts: log sequence clustering, label probability evaluation, and the structure and training of PLE-GRU. The first two steps entail creating pseudo-labels by utilizing annotated labels from a portion of normal log sequences within the training dataset.

### 3.3.1 Log sequence clustering

Based on the idea that log sequences that have similar meanings are expected to be assigned identical labels, PLE-GRU utilizes advanced clustering techniques to group log sequences with comparable meanings. In this study, we utilize HDBSCAN [32] to cluster both labeled and unlabeled log sequences within the training set. The reason for this choice is that HDBSCAN is a data clustering technique that does not necessitate predefining the cluster count, unlike approaches such as K-means, and it has fewer parameters and is robust to parameter settings. The implementation of log sequence clustering is achieved through the hdbscan (https://hdbscan.readthedocs.io/en/latest/) package.

### 3.3.2 Label probability estimation

Given the complexity of achieving perfect clustering results, PLE-GRU adopts a strategy of assigning probabilistic labels to unlabeled log sequences instead of deterministic ones. This method involves evaluating the probability that an unlabeled log sequence corresponds to each label, thereby reducing the impact of noise introduced during clustering. Specifically, we compute the probability of an unlabeled log sequence belonging to each label based on clustering outcomes. Using HDBSCAN, each log sequence in a cluster receives a score indicating the uncertainty of its cluster membership. This score, ranging from 0 to 1, serves as a measure of confidence in clustering the log sequence with its respective group; a lower score indicates higher confidence. Despite potential uncertainty, assigning a probabilistic label is crucial to align with the initial label estimation framework. By leveraging these principles and the scores from HDBSCAN clustering, each preliminary label is converted into a probabilistic label where $P$ (anomalous) = 1 − score/2 and $P$ (normal) = score/2.

### 3.3.3 The structure and training of PLE-GRU

The pseudo-labels derived from the training dataset by estimating label probabilities will be utilized for training a Gated Recurrent Unit (GRU) neural network, establishing a robust and efficient anomaly detection model. GRU is a type of recurrent neural network (RNN) architecture devised to combat the vanishing gradient problem encountered in traditional RNNs. It bears resemblance to LSTM but boasts a simpler structure featuring two primary gates: the update gate and the reset gate. GRU is renowned for its ability to capture long-term dependencies in sequential data effectively, requiring fewer parameters compared to LSTM.

For a log sequence represented as $S = \{v^1, v^2, \ldots, v^t\}$, where $v^t$ $(1 < t < T)$ denotes the $t$-th log event, and $T$ signifies the total log events in $S$, the input to the GRU at time step $t$ is the semantic vector of $v^t$ designated as $x_t$. The GRU cell computation involves two key elements: the hidden state ($h_t$) and the update gate ($z_t$) along with the reset gate ($r_t$). The computations for these components at time step $t$ are expressed by the following formulas:

TABLE 1 Statistics of HDFS and BGL.

| Dataset | HDFS | BGL |
|---|---|---|
| Event Collection/day | 2 | 215 |
| Size/GB | 1.490 | 0.708 |
| Number of Logs | 1,175,629 | 4,747,963 |
| Number of Anomalies | 16838 (blocks) | 348460 (logs) |
| Total Number of Templates | 30 | 378 |
| Number of Training Sequences | 5,000 | 7,500 |
| Number of Training Templates | 15 | 185 |

(1) Update Gate:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{12}$$

(2) Reset Gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{13}$$

(3) Candidate Hidden State:

$$\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t]) \tag{14}$$

(4) Update Hidden State:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{15}$$

At each time step $t$, $x_t$ denotes the input log event, $h_{t-1}$ stands for the hidden state from the preceding time step, and $W_z$, $W_r$, $W_h$ represent weight matrices. The function $\sigma$ signifies the sigmoid activation, with $*$ indicating element-wise multiplication. The ultimate hidden state is leveraged to predict whether the input log sequence is anomalous.

## 4 Experimental results and analysis

### 4.1 Dataset

To evaluate the performance of LogMS, experiments were carried out on the Hadoop Distributed File System (HDFS) dataset and the Blue Gene/L supercomputer (BGL) dataset, followed by a comprehensive analysis of the outcomes. These datasets are commonly utilized in log anomaly detection, with their characteristics outlined in Table 1. The HDFS dataset, generated by over 200 Amazon EC2 nodes, comprises a total of 11,175,629 log messages. These log entries are segmented into distinct log windows based on their corresponding block_id, representing the program execution status within the HDFS system. Among the log entries, 16,838 log blocks (2.93%) indicate system anomalies. On the other hand, the BGL dataset encompasses 4,747,963 log messages from the “Blue Gene/L” supercomputer, which houses 128 K processors at Lawrence Livermore National

TABLE 2 Comparison of the model structures of the baseline methods and LogMS.

| Models | Backbone | Label information | Semantic information | Sequential information | Quantitative information |
|---|---|---|---|---|---|
| DeepLog | LSTM | × | × | ✓ | × |
| LogAnomaly | LSTM | × | × | ✓ | ✓ |
| LogRobust | Bi-LSTM | ✓ | ✓ | × | × |
| Lu et al. | CNN | ✓ | × | × | × |
| LogMS | LSTM + GRU | ✓ | ✓ | ✓ | ✓ |

TABLE 3 The results of comparative experiments on HDFS and BGL.

| Models | HDFS | | | BGL | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| DeepLog | 0.945 | 0.899 | 0.922 | 0.900 | 0.960 | 0.929 |
| LogAnomaly | 0.860 | 0.897 | 0.877 | 0.970 | 0.940 | 0.960 |
| LogRobust | 0.961 | 0.999 | 0.980 | 0.994 | 0.942 | 0.967 |
| Lu et al. | 0.966 | 0.998 | 0.982 | 0.994 | 0.963 | 0.978 |
| LogMS | 0.997 | 0.998 | 0.998 | 0.994 | 0.987 | 0.984 |

TABLE 4 The results of ablation experiments on HDFS and BGL.

| Models | Metrics | HDFS | BGL |
|---|---|---|---|
| MSIF-LSTM | Precision | 0.865 | 0.970 |
| | Recall | 0.903 | 0.940 |
| | F1 | 0.882 | 0.960 |
| PLE-GRU | Precision | 0.950 | 0.965 |
| | Recall | 0.963 | 0.999 |
| | F1 | 0.957 | 0.982 |
| LogMS | Precision | 0.997 | 0.994 |
| | Recall | 0.998 | 0.987 |
| | F1 | 0.998 | 0.984 |

Laboratory. This dataset spans over 7 months, with experts in the BGL domain manually categorizing each log entry as abnormal or normal. Notably, there are 348,460 abnormal log messages in the BGL dataset. Unlike HDFS, the BGL dataset lacks explicit labels like block_id, making it challenging to extract log sequences effectively.

After log parsing, a total of 30 HDFS log templates and 378 BGL log templates are obtained. For HDFS, the logs are divided into sequences based on block_id. For BGL, as the logs do not record identifiers for each sequence, a fixed window size of 150 is used to segment the logs into sequences.

## 4.2 Evaluation metrics

In this study, precision, recall, and F1-score are employed as evaluation metrics, commonly utilized in log anomaly detection research [13–15]. Precision measures the proportion of accurately identified abnormal log sequences among all sequences flagged as anomalies by the model, calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

Recall gauges the proportion of correctly identified abnormal log sequences among all actual anomalies, expressed as:

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

F1-score, the harmonic mean of precision and recall, is calculated as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{18}$$

Here, TP (True Positive) indicates the count of abnormal log sequences correctly identified by the model, FP (False Positive) represents the number of normal log sequences inaccurately classified as anomalies, and FN (False Negative) denotes the count of abnormal log sequences overlooked by the model.

## 4.3 Experimental setting

We implement LogMS based on Python 3.8.3 and PyTorch 1.5.1. All experiments are conducted on a single RTX 3090Ti 24 GB GPU. In MSIF-LSTM, we set the weight decay to 0.0001, momentum to 0.9, initial learning rate to 0.01, use cross-entropy as the loss function, set the mini-batch size to 128, and train for 10 epochs. In PLE-GRU, we set the min_cluster_size parameter in HDBSCAN to 100, min_samples to 100, and train for 20 epochs.

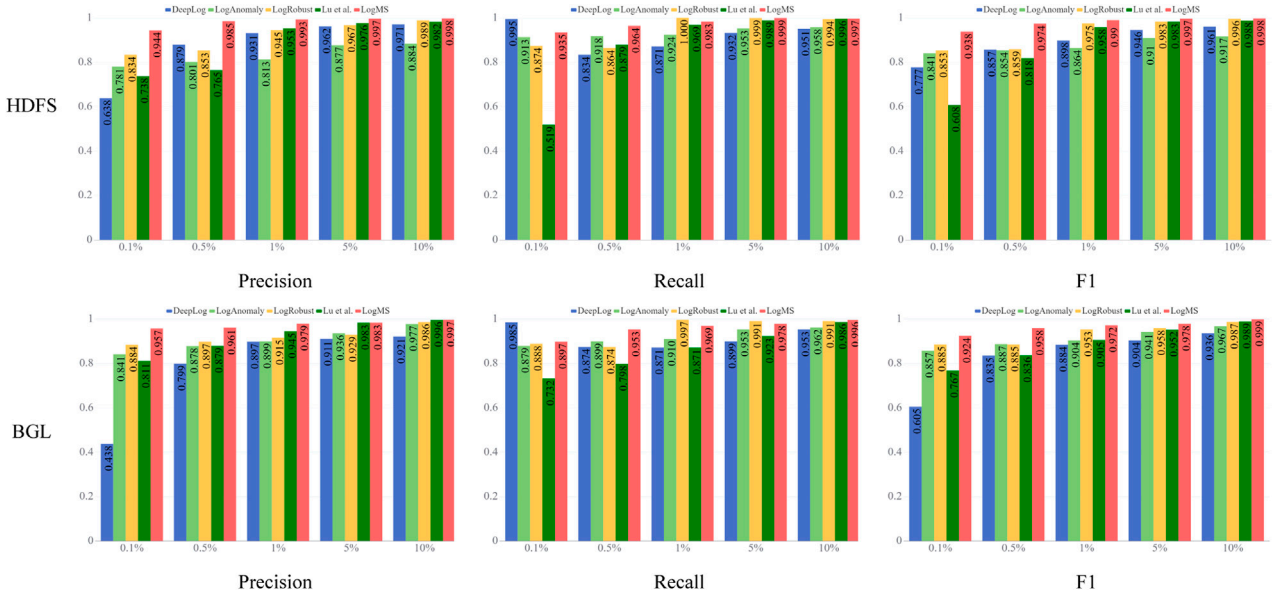



FIGURE 4
The results of class imbalance experiments in different methods.

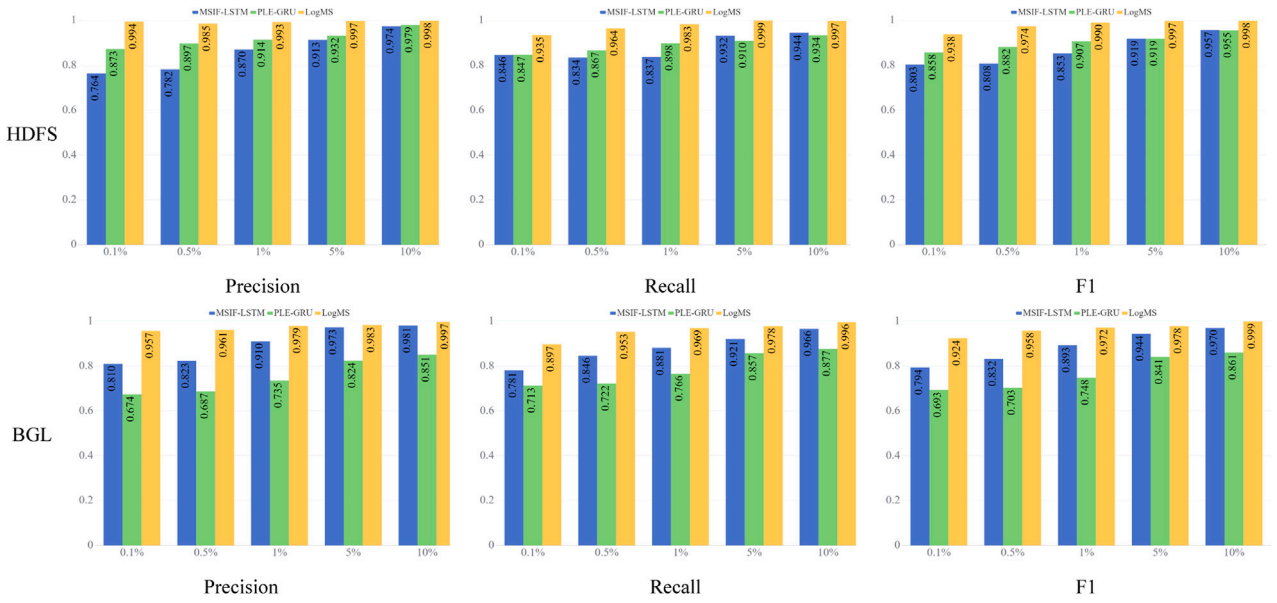



FIGURE 5
The results of class imbalance experiments in different component of LogMS.

## 4.4 Comparative experiments

We compare LogMS with the following four widely used methods, and the comparison of the model structures are shown in Table 2.

DeepLog [14]: This method treats system logs as natural language sequences and uses LSTM to model the sequential information of the logs.

LogAnomaly [15]: This method also treats system logs as natural language sequences, but uses LSTM to model sequential and quantitative information of the logs.

LogRobust [13]: This method is able to identify and handle unstable log events and sequences and uses Bi-LSTM to model label and sematic information.

Lu et al. [23]: This method can automatically learn event relationships in system logs and uses CNN to model label information.

Table 3 displays the results of comparative experiments. In HDFS, LogRobust achieves the highest F1 score, mainly due to its higher recall rate. LogMS performs second best, with a F1 score only 0.001 lower than LogRobust, but it has a higher precision by 0.17 compared to LogRobust. The worst performing models are DeepLog and LogAnomaly, mainly because they both not utilize label information and belong to unsupervised methods. Although LogRobust achieves the best results in HDFS, its performance in BGL is even worse than that of LogAnomaly, which does not utilize label information. Meanwhile, LogMS obtained the highest F1 score in BGL. Overall, LogMS performs well in both datasets and exhibited stable results. Compared to other methods, the key feature of LogMS lies in its effective fusion of multiple sources of information, demonstrating that the fusion of semantic, sequential, quantitative, and label information is an effective way to enhance the performance of log anomaly detection.

## 4.5 Ablation experiments

To assess the effectiveness of each improvement in LogMS, we conduct ablation experiments in HDFS and BGL. We divide LogMS into two parts: MSIF-LSTM, which integrates semantic, sequential, and quantitative information; PLE-GRU, which incorporates label information. LogMS fuses all four types of information. Table 4 presents the results of the ablation experiments.

Based on the experimental results, it is evident that MSIF-LSTM performs better in terms of precision, while PLE-GRU exhibits higher recall. LogMS combines the strengths of both, achieving the best precision and recall simultaneously. It is noteworthy that in the two-stage process of LogMS, MSIF-LSTM serves as the first stage, and only when MSIF-LSTM fails to detect anomalies, it proceeds to the second stage, PLE-GRU. The high precision of MSIF-LSTM in the first stage ensures a low false negative rate, while the high recall of PLE-GRU in the second stage minimizes missing anomalies, thus LogMS effectively integrates the strengths of both approaches.

## 4.6 Class imbalance experiments

A significant feature of log data is the substantial class distribution imbalance between normal logs and anomaly logs, as observed in datasets like HDFS where anomalies represent only about 2.9% of the data. Therefore, the ability of a model to deal with such situation is crucial [33]. In order to systematically assess our approach, we introduce various imbalanced scenarios by randomly excluding normal or abnormal log sequences from the HDFS and BGL dataset. We vary the imbalance ratio from 1% to 15%, indicating the percentage of anomalies present in the dataset. This process results in the creation of four synthetic datasets with imbalance ratios set at 0.1%, 0.5%, 1%, 5%, and 10%. To comprehensively evaluate our model, we conduct class imbalance experiments not only across different methods but also on the various components of LogMS. The experimental results are illustrated in Figures 4, 5.

From Figure 4, we can observe that as the proportion of abnormal labels increases, both precision and F1 improve, while recall remains stable. The reason for this phenomenon is as follows: due to the scarcity of positive samples, an increase in the number of positive samples results in an increase in true positives without a significant rise in false positives, leading to an enhancement in precision. However, recall is influenced by the imbalance in samples; when the number of positive samples is low, even with an increase in true positives, the number of false negatives may also rise, causing recall to be unstable and unable to consistently improve with an increase in positive samples. Despite the unstable recall, the improvement in precision leads to an overall increase in the F1 score. Overall, LogMS demonstrates robustness to severe class imbalance, particularly achieving optimal performance at the anomaly ratio of 0.1%.

We can see a similar phenomenon in Figure 5 as in Figure 4. However, the two-stage strategy of LogMS enables the effective integration of both components, thus maintaining the stability of log anomaly detection performance even under class imbalance conditions.

# 5 Conclusion

Deep learning-based log anomaly detection models primarily adopt a single-stage detection method and mainly focus on a specific aspect of log information. However, logs contain multiple sources of information (such as semantic information, sequential information, quantitative information, and label information). By focusing solely on a single aspect, the detection models are limited in their understanding of logs, resulting in compromised detection performance and suboptimal robustness. To address this issue, the paper introduces a multi-stage log anomaly detection method named LogMS. This method is based on the fusion of multiple sources of information (i.e., MSIF-LSTM) and probability label estimation (i.e., PLE-GRU), allowing for comprehensive utilization and fusion of various hidden information embedded in log data from multiple perspectives. Experimental results demonstrate that LogMS outperforms baseline models on various log anomaly detection datasets, demonstrating superior performance in robustness testing. In future research, we will consider integrating more sources of information such as system metrics, network traffic, or user behavior patterns to provide more comprehensive insights into log anomalies. By integrating these contextual factors into the detection process, it is possible to improve the accuracy and robustness of log anomaly detection models.

# Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/logpai/loghub.

# Author contributions

ZY: Data curation, Formal Analysis, Methodology, Writing–original draft. SY: Methodology, Supervision, Writing–review and editing. ZL: Data curation, Formal Analysis,

Writing–review and editing. LL: Investigation, Validation, Writing–review and editing. HL: Data curation, Formal Analysis, Writing–review and editing. FY: Formal Analysis, Investigation, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. Key Technology Project of China Tobacco Yunnan Industrial (2023ZN04). The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

## Conflict of interest

Authors ZY, SY, ZL, LL, HL, and FY were employed by China Tobacco Yunnan Industrial Co., Ltd.

## Publisher's note



## References

1. Landauer M, Onder S, Skopik F, Wurzenberger M. Deep learning for anomaly detection in log data: a survey. *Machine Learn Appl* (2023) 12:100470. doi:10.1016/j.mlwa.2023.100470

2. Chen Z, Liu J, Gu W, Su Y, Lyu MR. Experience report: deep learning-based system log analysis for anomaly detection. arXiv preprint arXiv:2107.05908 (2021).

3. Le V-H, Zhang H. Log-based anomaly detection without log parsing. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE (2021). p. 492–504.

4. Ko J, Comuzzi M. A systematic review of anomaly detection for business process event logs. *Business Inf Syst Eng* (2023) 65:441–62. doi:10.1007/s12599-023-00794-y

5. Guo H, Yuan S, Wu X. Logbert: log anomaly detection via bert. In: 2021 international joint conference on neural networks (IJCNN). IEEE (2021). p. 1–8.

6. Breier J, Brani sová J. Anomaly detection from log files using data mining techniques. *Inf Sci Appl* (2015) 339:449–57. doi:10.1007/978-3-662-46578-3_53

7. He S, Zhu J, He P, Lyu MR. Experience report: system log analysis for anomaly detection. In: 2016 IEEE 27th international symposium on software reliability engineering (ISSRE). IEEE (2016). p. 207–18.

8. Han D, Wang Z, Chen W, Wang K, Yu R, Wang S, et al. (2023). Anomaly detection in the open world: normality shift detection, explanation, and adaptation. In , doi:10.14722/ndss.2023.24830NDSS

9. Le V-H, Zhang H. Log-based anomaly detection with deep learning: how far are we? In: Proceedings of the 44th international conference on software engineering (2022). p. 1356–67. doi:10.1145/3510003.3510155

10. Nassif AB, Talib MA, Nasir Q, Dakalbab FM. Machine learning for anomaly detection: a systematic review. *Ieee Access* (2021) 9:78658–700. doi:10.1109/access.2021.3083060

11. Guo H, Guo Y, Yang J, Liu J, Li Z, Zheng T, et al. Loglg: weakly supervised log anomaly detection via log-event graph construction. In: International Conference on Database Systems for Advanced Applications. Springer (2023). p. 490–501.

12. Lee Y, Kim J, Kang P. Lanobert: system log anomaly detection based on bert masked language model. *Appl Soft Comput* (2023) 146:110689. doi:10.1016/j.asoc.2023.110689

13. Zhang X, Xu Y, Lin Q, Qiao B, Zhang H, Dang Y, et al. Keratin 6, 16 and 17-critical barrier alarmin molecules in skin wounds and psoriasis. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 8 (2019). p. 807–17. doi:10.3390/cells8080807

14. Du M, Li F, Zheng G, Srikumar V. Deeplog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (2017). p. 1285–98. doi:10.1145/3133956.3134015

15. Meng W, Liu Y, Zhu Y, Zhang S, Pei D, Liu Y, et al. Loganomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. *IJCAI* (2019) 19:4739–45. doi:10.24963/ijcai.2019/658

16. He P, Zhu J, Zheng Z, Lyu MR. Drain: an online log parsing approach with fixed depth tree. In: 2017 IEEE international conference on web services (ICWS). IEEE (2017). p. 33–40.

17. Xu W, Huang L, Fox A, Patterson D, Jordan M. Largescale system problem detection by mining console logs. In: Proceedings of SOSP'09 (2009).

18. Oliner A, Stearley J. What supercomputers say: a study of five system logs. In: 37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07). IEEE (2007). p. 575–84.

19. Wang H, Bah MJ, Hammad M. Progress in outlier detection techniques: a survey. *Ieee Access* (2019) 7:107964–8000. doi:10.1109/access.2019.2932769

20. Reidemeister T, Jiang M, Ward PA. Mining unstructured log files for recurrent fault diagnosis. In: 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops. IEEE (2011). p. 377–84.

21. Bai Y, Shu Z, Yu J, Yu Z, Wu X-J. Proxy-based graph convolutional hashing for cross-modal retrieval. *IEEE Trans Big Data* (2023) 1–15. doi:10.1109/tbdata.2023.3338951

22. Li L, Shu Z, Yu Z, Wu X-J. Robust online hashing with label semantic enhancement for cross-modal retrieval. *Pattern Recognition* (2024) 145:109972. doi:10.1016/j.patcog.2023.109972

23. Lu S, Wei X, Li Y, Wang L. Detecting anomaly in big data system logs using convolutional neural network. In: 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). IEEE (2018). p. 151–8.

24. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE* (1998) 86:2278–324. doi:10.1109/5.726791

25. Shu Z, Li B, Mao C, Gao S, Yu Z. Structure-guided feature and cluster contrastive learning for multi-view clustering. *Neurocomputing* (2024) 582:127555. doi:10.1016/j.neucom.2024.127555

26. Farzad A, Gulliver TA. Unsupervised log message anomaly detection. *ICT Express* (2020) 6:229–37. doi:10.1016/j.icte.2020.06.003

27. Du M, Li F. Spell: streaming parsing of system event logs. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE (2016). p. 859–64.

28. Yu S, He P, Chen N, Wu Y. Brain: log parsing with bidirectional parallel tree. *IEEE Trans Serv Comput* (2023) 16:3224–37. doi:10.1109/tsc.2023.3270566

29. Xu J, Yang R, Huo Y, Zhang C, He P. Divlog: log parsing with prompt enhanced in-context learning. In: 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE). IEEE Computer Society (2024). p. 983.

30. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016). Fasttext.zip: compressing text classification models. *arXiv preprint arXiv:1612.03651*

31. Shu Z, Li L, Yu J, Zhang D, Yu Z, Wu X-J. Online supervised collective matrix factorization hashing for cross-modal retrieval. *Appl intelligence* (2023) 53:14201–18. doi:10.1007/s10489-022-04189-6

32. McInnes L, Healy J, Astels S. hdbscan: hierarchical density based clustering. *J Open Source Softw* (2017) 2:205. doi:10.21105/joss.00205

33. Shu Z, Yong K, Yu J, Gao S, Mao C, Yu Z. Discrete asymmetric zero-shot hashing with application to cross-modal retrieval. *Neurocomputing* (2022) 511:366–79. doi:10.1016/j.neucom.2022.09.037

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 23 May 2024
DOI 10.3389/fphy.2024.1404448



OPEN ACCESS

EDITED BY
Zhenqiu Shu,
Kunming University of Science and Technology, China

REVIEWED BY
Jiaxu Leng,
Chongqing University of Posts and Telecommunications, China
Teng Sun,
Kunming University of Science and Technology, China

*CORRESPONDENCE
Zheng Xu,
zheng.xu@giat.ac.cn

†These authors share first authorship

RECEIVED 21 March 2024
ACCEPTED 29 April 2024
PUBLISHED 23 May 2024

CITATION
Chen J, Huang Z, Jiang X, Yuan H, Wang W, Wang J, Wang X and Xu Z (2024), Authenticity identification method for calligraphy regular script based on improved YOLOv7 algorithm. *Front. Phys.* 12:1404448.
doi: 10.3389/fphy.2024.1404448



# Authenticity identification method for calligraphy regular script based on improved YOLOv7 algorithm

Jinyuan Chen[1†], Zucheng Huang[1†], Xuyao Jiang[1], Hai Yuan[1], Weijun Wang[1,2], Jian Wang[1], Xintong Wang[1] and Zheng Xu[1*]

[1]Guangzhou Institute of Advanced Technology, Guangzhou, China, [2]Shenzhen Cas Derui Intelligent Technology Co., Ltd., Shenzhen, China

A regular calligraphy script of each calligrapher has unique strokes, and a script's authenticity can be identified by comparing them. Hence, this study introduces a method for identifying the authenticity of regular script calligraphy works based on the improved YOLOv7 algorithm. The proposed method evaluates the authenticity of calligraphy works by detecting and comparing the number of single-character features in each regular script calligraphy work. Specifically, first, we collected regular script calligraphy works from a well-known domestic calligrapher and divided each work into a single-character dataset. Then, we introduced the PConv module in FasterNet, the DyHead dynamic detection header network, and the MPDiou bounding box loss function to optimize the accuracy of the YOLOv7 algorithm. Thus, we constructed an improved algorithm named YOLOv7-PDM, which is used for regular script calligraphy identification. The proposed YOLOv7-PDM model was trained and tested using a prepared regular script single-character dataset. Through experimental results, we confirmed the practicality and feasibility of the proposed method and demonstrated that the YOLOv7-PDM algorithm model achieves 94.19% accuracy (mAP50) in detecting regular script font features, with a single-image detection time of 3.1 m and 31.67M parameters. The improved YOLOv7 algorithm model offers greater advantages in detection speed, accuracy, and model complexity compared to current mainstream detection algorithms. This demonstrates that the developed approach effectively extracts stroke features of regular script calligraphy and provides guidance for future studies on authenticity identification.

KEYWORDS

**calligraphy works identification, YOLOv7 algorithm, PConv module, DyHead dynamic detection head network, MPDiou loss function**

## 1 Introduction

Calligraphy, as a unique form of artistic expression, has a long history in China and stands out in the progression of human civilization [1]. Due to their significant collection value and potential for appreciation, calligraphy works are highly sought after by collectors both domestically and internationally, particularly those created by renowned ancient calligraphers [2]. However, genuine works by master calligraphers are becoming increasingly scarce, leading to abundant forgeries in the market. Consequently, there is an urgent need for calligraphy authenticity identification.

Traditional methods of calligraphy identification mainly involve three approaches [3]. One relies on experienced calligraphy experts with solid skills and substantial experience for empirical identification [4]. However, subjective factors often influence this method, biasing the identification results. An alternative approach utilizes physical techniques to determine authenticity by examining the presence of seals and analyzing the composition of paper used in the calligraphy work. Nevertheless, as technology advances, forgery techniques have become increasingly sophisticated, with the ability to replicate seals and paper, resulting in identification biases [5]. The third method uses computer-assisted techniques to detect the authenticity of calligraphic works. With the further development of computer science and technology in recent years, many researchers have employed computer-assisted methods to detect the authenticity of calligraphic works. However, computer-assisted methods can be further categorized into two types: one is based on traditional image processing algorithms, such as the calligraphic work authentication method proposed by Zeng [6] based on image recognition and the computer-assisted calligraphy authenticity identification proposed by Pang [7]. The other type employs novel image processing methods based on deep learning, such as Li's [8] evaluation and detection of calligraphic copying based on deep learning.

To address the challenge of the identification bias, this study develops an authenticity identification method for calligraphy regular script based on an improved YOLOv7 algorithm. Specifically, first, we manually annotate the features of individual characters in authentic calligraphy regular script works, followed by feature extraction using deep learning networks. The authenticity of calligraphy works is determined by comparing the number of extracted features from genuine works with the forged ones. This method aims to enhance the accuracy and reliability of calligraphy regular script authenticity identification by combining manual annotation and deep learning techniques.

The traditional algorithmic approach involves image processing, and after conducting feature extraction on the works of a single calligrapher, this approach exhibits relatively high detection accuracy. However, the detection algorithm cannot be directly applied to the works of another calligrapher, thus posing significant limitations. Unlike simplistic image processing schemes, deep learning can automatically learn features and exhibits strong robustness and adaptability, enabling accurate detection and recognition in complex environments. Furthermore, deep learning approaches demonstrate high generalization and are suitable for detecting the works of most calligraphers using the same font style [9]. Deep learning has experienced extensive application and has recently advanced significantly in diverse domains. For instance, Wang [10] employed an improved EfficientNet algorithm to authenticate calligraphic works, efficiently categorizing genuine from fake calligraphic pieces using the two-class classification property of the EfficientNet algorithm. The corresponding experimental results demonstrated significant effectiveness. Xu [11] proposed an improved YOLOv4-Tiny algorithm that effectively detects boats on rivers and lakes, ensuring waterway safety. Hu [12] applied the improved YOLOX algorithm to rapidly detect surface hole defects on aluminum castings, enhancing casting efficiency. Mai made a breakthrough in calligraphy font recognition using DenseNet networks [13]. The advantages of deep learning methods lie in their ability to learn features automatically and possess strong robustness and adaptability in accurate detection and recognition in complex environments. Therefore, utilizing deep learning methods for calligraphy regular script authenticity identification holds great potential and feasibility. Hence, building upon these successful research achievements, we leverage deep learning methods to authenticate calligraphy regular script works. Indeed, by constructing a deep learning model suitable for regular calligraphy script works, we extract and analyze the features of each character and compare these features with those of authentic works to determine the degree of authenticity. However, additional datasets and annotations may be required for training and validating the algorithm model. The proposed authenticity identification method is based on the improved YOLOv7 algorithm evaluating the authenticity of regular calligraphy scripts by detecting and comparing the features in each character.

# 2 Calligraphy regular script stroke feature detection algorithm based on YOLOv7-PDM

## 2.1 YOLOv7 algorithm

The YOLOv7 algorithm [14], introduced by the YOLOv4 [15] team, is another significant breakthrough in the YOLO series. Since its proposal at the end of 2022, the YOLOv7 algorithm has received considerable attention from the academic community, as it demonstrates excellent performance with a detection speed ranging from 5 to 160 FPS and exhibits higher detection accuracy and speed levels than current mainstream object detection algorithms. Figure 1 illustrates the structure of the YOLOv7 model [16].

The YOLOv7 algorithm comprises four main components: Input, the feature extraction network known as Backbone, the feature fusion network identified as Neck, and the detection head network referred to as YOLO-Head. Compared to prior YOLO algorithms, YOLOv7 presents innovative improvements in its Backbone, Neck, and YOLO head. The feature extraction network comprises CBS, ELAN, and MP1 convolution modules. The CBS module is a conventional convolution module consisting of regularization and activation functions, whereas the ELAN module is a layer aggregation network that improves efficiency. Additionally, dilation and transformation methods are used to enhance the learning performance of the algorithm model, boosting the model's computational capability while maintaining the original gradient path intact. The MP1 convolution module is formed by adding a Maxpool layer after the CBS module, which forms two branches combined with a Concat module to integrate the characteristics of both branches and enhance the network's ability to extract features. YOLOv7 has modified the SPP module in the Neck to the SPPCSPC module, a revised adaptation of Spatial Pyramid Pooling, to accommodate inputs of varying sizes. This modification reduces the image distortion caused by image processing and overcomes the feature re-extraction problem during convolution. In 2021, Megvii Technology published the PAFPN model, which incorporates the same feature pyramid network structure as YOLOX. Feature fusion between layers is

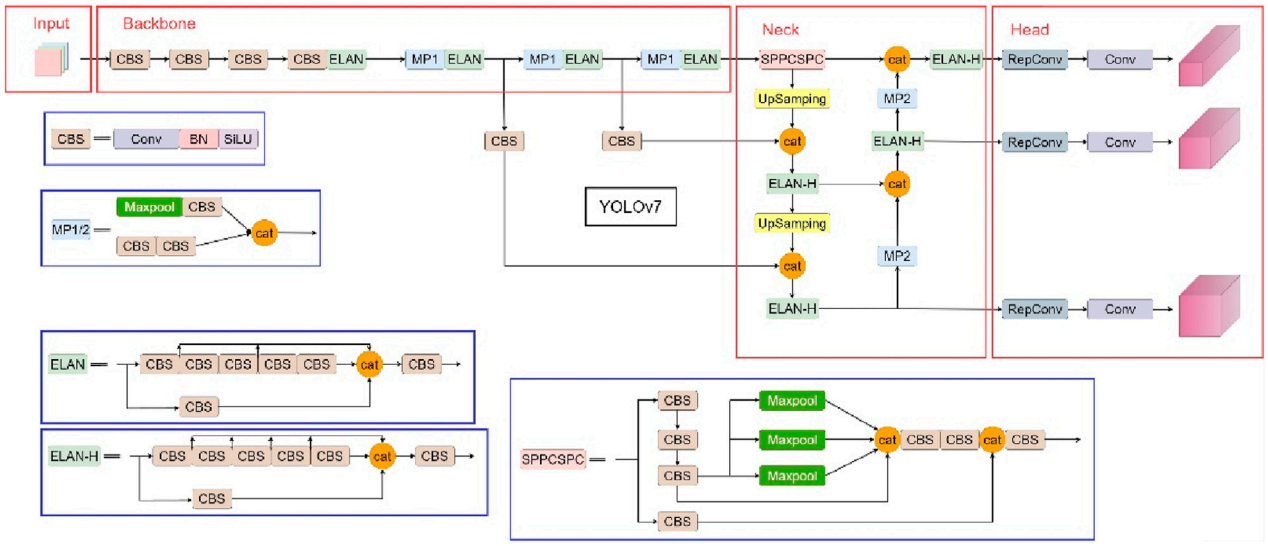



FIGURE 1
Structure of the YOLOv7 algorithm model.

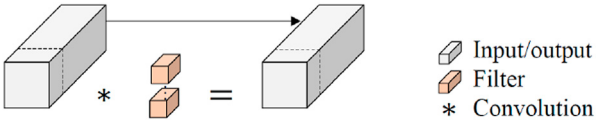



FIGURE 2
Structure of the PConv module.

achieved by passing deep features from bottom to top. Additionally, the Neck network includes the ELAN-H and MP2 modules, where the ELAN-H module aggregates more layers than the ELAN module. The only variation between the MP1 and MP2 modules is the number of channels. In the YOLO-Head, YOLOv7 combines the RepConv module's re-parameterized convolutions with the network structure, balancing speed and accuracy during training.

## 2.2 PConv module

To enhance the detection accuracy of the YOLOv7 algorithm, we replace the Conv layer in the CBS module with the PConv module from FasterNet [17]. The modified module has been renamed the PBS module. The PConv module plays a vital role in FasterNet, a novel image classification algorithm introduced in CVPR2023, which attains an exceptional TOP-1 accuracy of 83.5% on ImageNet-1k. The structure of the PConv module is illustrated in Figure 2.

PConv addresses higher memory access and reduces the overall computational complexity caused by depthwise separable convolution (DWConv), particularly on I/O-bound devices. DWConv can reduce the computational complexity of Conv by a factor of (number of channels), but the detection accuracy decreases as a result of the cost incurred. To mitigate the accuracy loss, the channel width must be increased to compensate for the decrease in parameter quantity. However, when DWConv is applied with an increased channel width, it introduces higher memory access and generates more computational redundancy. Considering these limitations, PConv performs regular Convolution on a specific group of input channels to extract spatial features while keeping the rest unaltered. The first or last consecutive channels represent the entire feature map for computation with consecutive or regular memory access. Without any loss of generality, it is assumed that the input and output feature maps have the same number of channels. Therefore, Eq. 1 defines the FLOPs of PConv, while Eq. 2 depicts the memory access.

$$FLOPs = h \times w \times k^2 \times c_p^2, \tag{1}$$

$$h \times w \times 2c_p + k^2 \times c_p^2 \approx \mathrm{h} \times w \times 2c_p, \tag{2}$$

In this case, the width and height of the feature map are represented by h and w, respectively. The size of the convolution kernel is denoted by k, and $c_p$ indicates the number of channels affected by regular convolution. This cp value is equivalent to the change from $c_{in}$ to $c_p$ in conventional convolution. However, in practical scenarios, PConv uses only one-fourth of the channels present in $c_p$ which leads to a reduction of FLOPs by 1/16 and memory access by 1/4 compared to conventional convolution.

## 2.3 DyHead dynamic detection head

The DyHead dynamic detection head network proposed by Microsoft [18] aims to enhance the detection accuracy of the YOLOv7 algorithm. DyHead is a dynamic detection network that introduces attention mechanisms to consolidate different object detection heads innovatively. The core idea of this method is to leverage attention mechanisms to enable interaction among scales (referred to as $\pi_L$), spatial (referred to as $\pi_S$), and task (referred to as $\pi_C$) awareness based on a given feature tensor, denoted as $F \in R^{L \times S \times C}$. Specifically, the $\pi_L$ attention mechanism facilitates

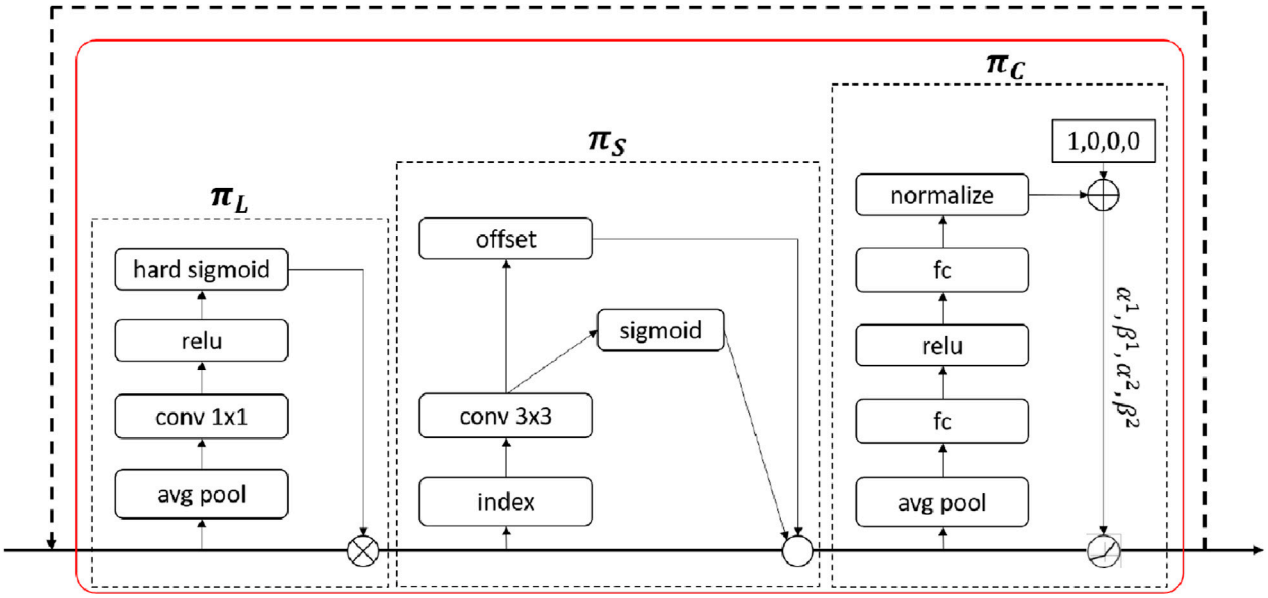



FIGURE 3
Structure of the DyHead module.

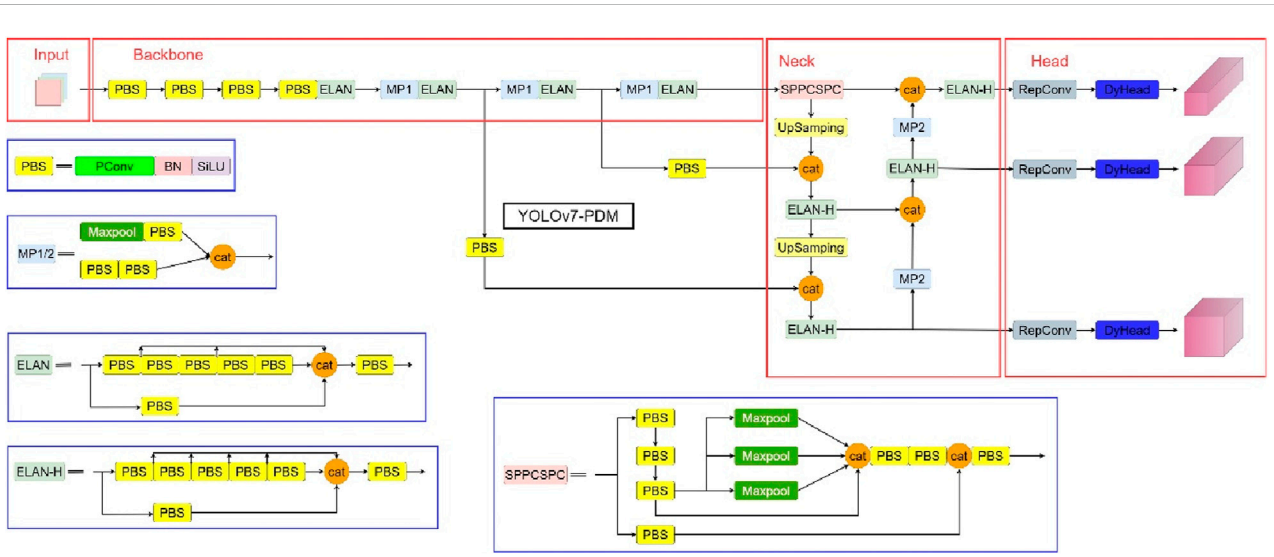



FIGURE 4
Model structure diagram of YOLOv7-PDM.

scale awareness between different feature levels, the $\pi_S$ attention mechanism enables spatial awareness between spatial positions and the $\pi_C$ attention mechanism promotes task awareness within the output channels. These $\pi_L$, $\pi_S$, and $\pi_C$ attention mechanisms are combined to form the DyHead dynamic detection head module, as illustrated in Figure 3. By introducing the DyHead dynamic detection head module, we effectively utilize attention mechanisms to improve the performance and accuracy of object detection. The novelty of this method lies in applying attention mechanisms to different levels of perception and achieving a unified object detection head network through modular design.

The general form of self-attention is presented by Eq. 3.

$$W(F) = \pi(F) \cdot F, \tag{3}$$

This form has many parameters and directly learns the attention function through fully connected layers across all dimensions. In order to enhance efficiency and reduce the number of parameters, we transformed this attention function into three separate attentions, each concentrating on a specific dimension, as presented in Eq. 4.

$$W(F) = \pi_C(\pi_S(\pi_L(F) \cdot F) \cdot F) \cdot F, \tag{4}$$

where $\pi_L$ combines the characteristics from various scales while considering their semantic significance, $\pi_S$ focuses on the discriminative capacity between different spatial positions and $\pi_C$

promotes joint learning and generalizability of target representation by dynamically switching feature channels to assist different tasks. Stacking the DyHead dynamic detection head module multiple times yields better performance improvement, which peaks after stacking more than six modules. By introducing the DyHead dynamic detection head module, the expressiveness of the YOLOv7 algorithm's YOLO-Head is significantly enhanced without substantially increasing the computational complexity.

## 2.4 MPDioU bounding box loss function

As an improvement, we introduce the MPDIoU bounding box loss function [19] to address the instability in expressing the aspect ratio penalty of the CIoU loss function when the aspect ratio of the predicted bounding box matches that of the ground truth bounding box in the original YOLOv7 algorithm. The latter bounding box initially utilizes the CIoU loss function for bounding box regression. The proposed MPDioU bounding box loss function, which relies on the minimum point distance, assesses the similarity between predicted and ground truth bounding boxes, acting as a criterion for comparison. It should be noted that the YOLOv7 algorithm's convergence speed and detection accuracy are constrained because the CIoU and EIoU lose their effectiveness when the predicted and ground truth bounding boxes have varying width and height values but the same aspect ratio. This issue is overcome by combining the benefits of CIoU and EIoU. Besides, MPDIoU takes inspiration from the geometric characteristics of bounding boxes by directly minimizing the distances between the top left and bottom right points of the predicted and ground truth bounding boxes. The specific implementation is presented in Eq. 5.

$$MPDioU = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2}, \tag{5}$$

where A and B are two bounding boxes, $d_1$ is the distance between their top left points, $d_2$ is the distance between their bottom right points, and w and h represent the width and height of the input image. This design simplifies the similarity comparison between two bounding boxes and applies to overlapping and non-overlapping bounding box regression. Consequently, by leveraging the benefits of the MPDIoU bounding box loss function, the accuracy of the YOLOv7 algorithm in detecting objects is enhanced.

## 2.5 YOLOv7-PDM algorithm model

Figure 4 overviews the structure diagram of the YOLOv7-PDM algorithm, which has been optimized by incorporating the PConv module, DyHead dynamic detection head, and MPDioU bounding box loss function.

# 3 Experiment

## 3.1 Dataset creation

Due to a shortage of publicly accessible datasets for regular script characters in calligraphy, this research meticulously compiled an exclusive dataset by utilizing genuine works from Shen, a renowned calligrapher and member of the China Calligraphers Association, provided by the Sanpin Art Gallery in Shenzhen City. Regular script characters in calligraphic works typically exhibit single color and high contrast characteristics, with most presenting a consistent and neat writing style. Therefore, the works were first scanned using a line-scan camera in the data preprocessing stage. Subsequently, traditional binarization techniques effectively separated the acquired images into foreground and background. Additionally, we obtained individual regular script characters by batch cropping, utilizing fixed spacing between the characters. As a result, 2,782 black-and-white image samples of regular script characters were obtained, as depicted in Figure 5.

Following the research on Chinese digital calligraphy retrieval and authenticity identification by Zhang et al. [20], the stroke features of regular script characters were categorized into three basic features: start (qi), turn (zhuan), and end (shou). The start and end features were further divided into horizontal start (hengqi), vertical start (shuqi), horizontal end (hengshou), and vertical end (shushou). The turning feature was classified into a right-angle turn (zhijiaoze) and an acute-angle turn (ruijiaoze). Therefore, six-stroke features were extracted from regular script characters. After obtaining the images of regular script characters, we annotated them using the DLtools (MVTec Deep Learning Tool) annotation software. The annotation process requires careful alignment with every feature of regular script calligraphy characters. Besides, the selection of feature boxes should be neither too large nor too small, and it is necessary to conduct repeated inspections to ensure the absence of missed annotations, as omitting a single feature could potentially impact the accuracy of subsequent model training. The specific annotation quality is illustrated in Figure 6, representing a favorable annotation standard. Among them are 5,343 characters with a horizontal starting stroke, 4,545 with a horizontal ending stroke, 7,542 with a vertical starting stroke, 3,991 with a vertical ending stroke, 1,658 with right-angle turns, and 3,074 with acute-angle turns. The number of characters with right-angle and acute-angle turns is small, as not every character contains these types of turns.

Each calligrapher's characters exhibit a unique style, with the most distinctive characteristics being evident in the three fundamental aspects of "start," "turn," and "end," Where "start" refers to the starting point of the stroke, signifying the moment the brush touches the paper. The pressure and angle of initiation vary among calligraphers. Additionally, "turn" involves the rotation of the brush, with some characters requiring a subtle adjustment while others may demand a more pronounced rotation. Finally, "end" marks the stroke's conclusion, representing the character's completion. Some calligraphers execute the termination process, while others incorporate personal stylistic elements to showcase individuality. Therefore, using these six brushstroke features can effectively encapsulate the unique stylistic characteristics of a calligrapher's regular script.

In order to effectively mitigate the overfitting phenomenon during the algorithm model training process, this study employed data augmentation techniques on the 2,782 black-and-white images of individual regular script characters, including spatial transformation methods and noise addition methods, to expand


FIGURE 5
Segmented grayscale images of regular script characters.

the dataset. Through these methods, the original images of individual regular script characters were augmented to a total of 5,687 images, significantly enlarging the dataset. Data augmentation not only significantly enhanced the generalization capability of the algorithm model but also optimized the training performance of the model. Concurrently, following the format required by the YOLOv7 algorithm for training datasets, this study meticulously constructed the dataset of individual regular script characters for calligraphy. In order to ensure the scientific and practical validity of the dataset, we rigorously divided the dataset into training, validation, and testing sets in a 7:2:1 ratio to guarantee the reliability and effectiveness of model evaluation. Through the comprehensive implementation of the steps mentioned above, the construction of the dataset of individual regular script characters for calligraphy has been completed, providing a solid data foundation for the subsequent training and evaluation of algorithm models.

## 3.2 Experimental setup

The experimental setup for this research involved an Intel i9-13900K CPU, 128 GB of RAM, and two NVIDIA RTX4090 GPU cards with 24 GB of VRAM each. We set up the appropriate operating system (Ubuntu 20.04), Python 3.9, CUDA 11.8, PyTorch 2.0.0, and related dependencies on the training machine to conduct training and simulation experiments. By utilizing such hardware configuration and software environment, we ensured the smooth progress of the experiments and obtained accurate and reliable results. Furthermore, these configurations provided sufficient computational resources and performance to support the training and evaluation.

## 3.3 Training parameters and evaluation metrics

Before training the model, it is necessary to set the evaluation metrics and initialize the training parameters. This study employed four metrics to evaluate the model's performance: Mean Average Precision (mAP) with an IoU of 0.5, detection speed per image, parameter quantity, and computational complexity (FLOPs). The evaluation metrics were selected based on a comprehensive algorithm performance and efficiency consideration. Specifically, evaluating the accuracy and precision of the object detection algorithm relies on using mAP with an IoU of 0.5, while the detection speed per image measures the algorithm's efficiency in processing. Additionally, the complexity and computational requirements of the model are indicated by the parameter quantity and computational complexity (FLOPs). The selection of these four evaluation metrics is based on the fact that the authenticity detection of regular script characters only pursues

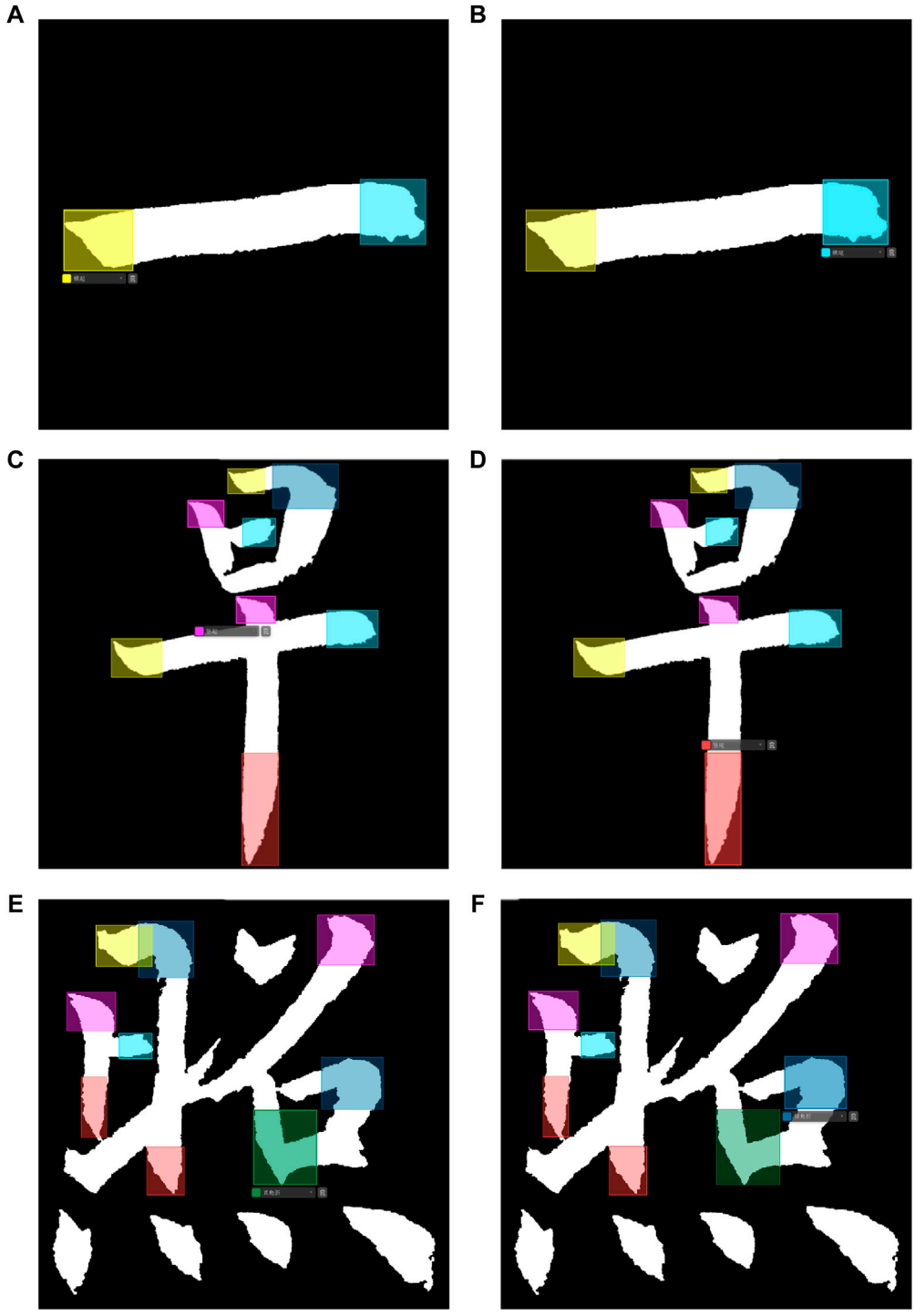



FIGURE 6
Six calligraphic character stroke types with their characteristic classification diagram highlighted: **(A)** horizontal start (yellow); **(B)** horizontal end (light blue); **(C)** vertical start (purple); **(D)** vertical end (red); **(E)** right angle turn (green); **(F)** acute angle turn (blue).

detection accuracy. Thus, the choice of these four evaluation metrics already satisfies the requirements.

Table 1 reports the precise configurations used to initialize the training parameters. Setting the hyperparameters is an important task impacting the model's performance and effectiveness. Moreover, ensuring hyperparameter setting consistency is crucial for enhancing the YOLOv7 algorithm model. On the one hand, preserving consistency in hyperparameter settings ensures effective algorithmic improvements while maintaining consistent hyperparameters, allowing for accurate evaluation of the

TABLE 1 Training parameters for the algorithm model.

| Parameter | Value |
|---|---|
| Initial Learning Rate(Init_Lr) | 0.02 |
| Minimum Learning Rate(Min_Lr) | 0.0002 |
| Total Training Epochs (Total_Epochs) | 1,000 |
| Learning Rate Decay Type (Lr_Decay_Type) | cos |
| Batch Size of Each Training (Batch_Size) | 48 |
| Optimizer Type of Network Architecture (Optimzer_Type) | SGD |
| Momentum of Optimization Function (Momentum) | 0.937 |
| Weight Decay Coefficient (Weight_Decay) | 0.0002 |

enhancements' effectiveness by comparing the algorithm's performance before and after the improvements. However, it is challenging to differentiate between the improvement effect of the algorithm itself and the performance changes caused by modifications to the hyperparameters when adjustments are made to the hyperparameters during the improvement process. Hence, the proposed method adopts the hyperparameters of YOLOv7.

In order to prevent overfitting of the regular script character dataset during the YOLOv7 algorithm training process, we measured the loss values of both the validation and training sets. After analysis, we found that the training set had a loss value (Loss) of 0.03, while the validation set had a loss value of 0.026, resulting in a minimal difference of only 0.004. This small difference suggests the absence of overfitting.

# 4 Experimental results

## 4.1 YOLOv7 algorithm with PConv module

We enhanced the Backbone and Neck sections of the YOLOv7 algorithm while considering the attributes of PConv. Specifically, the convolutions with a kernel size of 3 × 3 in the three feature output layers were replaced with PBS modules. The modified algorithm models in the SPPCSPC module, ELAN-H module, and the improved MP2 module in the Neck were labeled as YOLOv7-P, respectively. In order to guarantee the reliability of the experiments, this study conducted no less than 10 repeated experimental verifications on the YOLOv7 algorithm and YOLOv7-P algorithm on the proposed dataset. Table 2 reports the experimental results obtained by calculating the average of the experimental values when excluding the best and worst outcomes.

The experimental results in Table 2 highlight that the YOLOv7-P algorithm demonstrated a performance increase of almost 2.5% in mAP0.5 compared to the YOLOv7 algorithm. Additionally, the YOLOv7-P algorithm reduced the parameter quantity by 4.5M and FLOPs by one-fifth. Moreover, the single detection time remained almost unchanged between the two algorithms. By incorporating the PConv module into the YOLOv7 algorithm, the experimental results present enhanced detection accuracy and reduce the model's parameter quantity and computational complexity. This demonstrates the positive impact of the PConv module in the YOLOv7 algorithm without affecting the single detection time.

## 4.2 YOLOv7 algorithm with DyHead dynamic detection head

In order to evaluate the performance of integrating the DyHead dynamic detection head into the YOLOv7 algorithm (referred to as YOLO-Head) and to determine the optimal number of layers to embed the DyHead module, this study conducted no less than 10 repeated experimental verifications on the YOLOv7 algorithm and YOLOv7-D algorithm. To guarantee the reliability of the experiments, the experimental results were obtained by excluding the best and worst outcomes and averaging the remaining values. The detailed experimental results are presented in Table 3.

Table 3 infers that including four DyHead modules in the YOLOv7-D algorithm results in a performance enhancement of around 3.1% in mAP0.5 compared to the YOLOv7 algorithm. Additionally, the parameter quantity of the YOLOv7-D algorithm increases by 13M, while the FLOPs computational load shows a slight decrease. Furthermore, the detection time per image remains almost unchanged between the two algorithms. These experimental results demonstrate that although including the DyHead dynamic detection head in the YOLO-Head of the YOLOv7 algorithm leads to a relatively significant increase in parameter quantity, the FLOPs' computational load and detection time per image experience have insignificant changes. Moreover, the YOLOv7-D algorithm exhibits certain improvements in detection accuracy compared to the YOLOv7 algorithm. Thus, these findings substantiate the efficacy of integrating the DyHead dynamic detection head with the YOLOv7 algorithm.

## 4.3 YOLOv7 algorithm with MPDioU boundary box loss function

To assess the impact of replacing the CIoU bounding box loss function with the MPDioU bounding box loss function on the YOLOv7 algorithm's performance, we compared the training

TABLE 2 Experimental verification of PConv module.

| Algorithm | mAP0.5 (%) | Parameter quantity(M) | FLOPs(G) | Detection time per Image (ms) |
|---|---|---|---|---|
| YOLOv7 | 90.19 | 36.50 | 105.20 | 3.1 |
| YOLOv7-P | 92.53 | 32.00 | 82.96 | 3.2 |

TABLE 3 Experimental verification of DyHead dynamic detection head.

| Algorithm | Number of DyHead | mAP0.5 (%) | Parameter quantity(M) | FLOPs(G) | Detection time per Image (ms) |
|---|---|---|---|---|---|
| YOLOv7 | \ | 90.19 | 36.50 | 105.20 | 3.1 |
| YOLOv7-D | 1 | 90.88 | 39.27 | 84.73 | 3.1 |
| | 2 | 91.72 | 42.44 | 91.19 | 3.1 |
| | 3 | 92.53 | 46.03 | 97.65 | 3.2 |
| | 4 | 93.21 | 49.51 | 104.56 | 3.3 |
| | 5 | 93.03 | 54.14 | 110.68 | 3.4 |
| | 6 | 92.88 | 57.82 | 118.92 | 3.4 |

TABLE 4 Experimental verification of MPDioU boundary box loss function.

| Algorithm | mAP0.5 (%) | Parameter quantity(M) | FLOPs(G) | Detection time per Image (ms) |
|---|---|---|---|---|
| YOLOv7 | 90.19 | 36.50 | 105.20 | 3.1 |
| YOLOv7-M | 92.85 | 37.22 | 105.20 | 3.4 |

losses of both the regular YOLOv7 and the modified YOLOv7-M algorithms. The results reveal that the training loss of the YOLOv7-M algorithm is 0.02, while the training loss of the YOLOv7 algorithm is 0.03. This indicates that the MPDioU bounding box loss function is superior to the CIoU bounding box loss function. Furthermore, to ensure the validity of the experiments, we conducted no less than 10 repeated experiments on the YOLOv7 algorithm and the YOLOv7-M algorithm using the developed dataset. We calculated the average of the remaining experimental values after excluding the best and worst results to obtain the experimental results, with Table 4 presenting the experimental results.

Table 4 reveals that the YOLOv7-M algorithm achieves a boost of approximately 2.7% in mAP0.5 compared to the YOLOv7 algorithm. Additionally, the parameter quantity of the YOLOv7-M algorithm increases by nearly 1M, but there is no change in the FLOPs computational complexity, while the single detection time slightly increased. Considering these results, the YOLOv7-M algorithm model has higher detection accuracy under almost unchanged FLOPs computational complexity and single detection time. These results prove that the MPDioU boundary box loss function significantly enhances the performance of the YOLOv7 algorithm model.

## 4.4 Overall experiment analysis

### 4.4.1 Ablation experiment

This paper proposes three improvement methods, namely, the PConv module (YOLOv7-P), the DyHead dynamic detection head (YOLOv7-D), and the MPDioU bounding box loss function (YOLOv7-M). To ascertain the efficacy and enhancements of these three methods, comparative experiments were carried out under the same experimental settings to evaluate the performance disparities between the YOLOv7 algorithm and the YOLOv7 algorithm equipped with one, two, and three enhancement methods. To guarantee the experiments' validity, we repeated each experiment 10 times and excluded the most extreme results. The remaining values from the experiments were averaged to obtain the experimental outcome, as presented in Table 5.

Based on the findings in Table 5, the YOLOv7-PDM algorithm exhibits a 4% enhancement in mAP0.5 compared to the YOLOv7 algorithm. Furthermore, the YOLOv7-PDM algorithm has nearly 5M fewer parameters and approximately 27G FLOPs while maintaining the same detection time for individual images. These results suggest that the YOLOv7-PDM algorithm model surpasses the YOLOv7 algorithm model, considering operational and spatial complexity. Besides, the YOLOv7-PDM algorithm model, which integrates three enhancement methods, exhibits the highest performance, as it enhances detection accuracy (mAP0.5 improvement) and significantly reduces the parameter count and computational workload without impacting the time required for single-image detection.

### 4.4.2 Comparison with other mainstream object detection models

In order to assess the effectiveness of the YOLOv7-PDM algorithm, we carried out comparative experiments involving eight popular detection models: YOLOv7, YOLOv6 [21], YOLOv8 [22], Deformable-DETR [23], RT-DETR [24], Faster-RCNN [25], SSD [26], and DETR [27] under the same experimental configuration. To ascertain the experiment's validity, a minimum of 10 repetitions of experimental training and validation were conducted on all data results. The optimal and worst outcomes were disregarded, and the remaining experimental values were averaged to derive the final result. The experimental results are reported in Table 6.

TABLE 5 Ablation experiment comparison of three improvement methods.

| Algorithm | mAP0.5 (%) | Parameter quantity(M) | FLOPs(G) | Detection time per Image (ms) |
|---|---|---|---|---|
| YOLOv7 | 90.19 | 36.50 | 105.20 | 3.1 |
| YOLOv7-P | 92.53 | 32.00 | 82.96 | 3.2 |
| YOLOv7-D | 93.21 | 49.51 | 105.22 | 3.3 |
| YOLOv7-M | 92.85 | 37.22 | 105.20 | 3.4 |
| YOLOv7-PM | 92.78 | 32.00 | 82.96 | 2.4 |
| YOLOv7-PD | 93.62 | 31.67 | 78.20 | 4.3 |
| YOLOv7-DM | 93.47 | 36.18 | 98.47 | 3.6 |
| YOLOv7-PDM | 94.19 | 31.67 | 78.20 | 3.1 |

TABLE 6 Performance comparison of nine detection models.

| Algorithm | mAP0.5 (%) | Parameter quantity(M) | FLOPs(G) | Detection time per Image (ms) |
|---|---|---|---|---|
| YOLOv7 | 90.19 | 36.50 | 105.20 | 3.1 |
| YOLOv6 | 89.87 | 34.80 | 85.64 | 3.4 |
| YOLOv8 | 89.72 | 25.80 | 78.70 | 4.0 |
| DETR | 88.37 | 36.74 | 223.62 | 35.7 |
| Deformable-DETR | 87.98 | 39.83 | 157.35 | 36.2 |
| RT-DETR | 89.89 | 32.00 | 110.53 | 13.3 |
| Faster-RCNN | 85.33 | 41.38 | 269.03 | 27.3 |
| SSD | 53.94 | 13.69 | 30.71 | 1.3 |
| YOLOv7-PDM | 94.19 | 31.67 | 78.20 | 3.1 |

According to the results in Table 6, the YOLOv7-PDM algorithm outperforms the other eight mainstream algorithms in terms of mAP0.5, achieving 94.19%. The YOLOv7-PDM performs better in detection accuracy, showing a nearly 41% improvement in mAP0.5 while having fewer parameters, computational FLOPs, and detection time per image than the SSD algorithm. This indicates a significant advantage for YOLOv7-PDM. Compared with the Faster-RCNN algorithm, the YOLOv7-PDM outperforms it in all aspects, including mAP0.5, the number of parameters, computational FLOPs, and detection time per image. Moreover, relative to other YOLO series models, the YOLOv7-PDM achieves the highest levels of performance in mAP0.5, FLOPs, and detection time per image, with a slight disadvantage in the number of parameters compared to YOLOv8 but superior to YOLOv6. Compared with the DETR series models, the YOLOv7-PDM performs better in mAP0.5, the number of parameters, computational FLOPs, and detection time per image, validating the superiority of the proposed YOLOv7-PDM. In the detection of regular script characters, detection accuracy is more critical. Compared to the YOLOv7 algorithm, the YOLOv7-PDM algorithm maintains the same single-image detection time but substantially improves detection accuracy, parameter quantity, and computational FLOPs. This further validates the superiority of the proposed YOLOv7-PDM algorithm model in this study. Figure 7 illustrates the detection results of the nine models.

Comparing the graphs in Figure 7 reveals that when the pen stroke feature is small, the eight algorithm models fail to detect it correctly. It should be noted that in this paper, the size of the target is defined as follows, taking the commonly used dataset COCO object definition in the field of object detection as an example: small targets refer to objects smaller than 32 × 32 pixels, medium targets refer to objects ranging from 32 × 32 to 96 × 96 pixels, and large targets refer to objects larger than 96 × 96 pixels. When a single character has many strokes, leading to smaller pen stroke features, the SSD algorithm model fails to detect it. When there is a partial overlap in the pen stroke features, the YOLOv6, Faster-RCNN, and DETR algorithm models fail to detect it accurately. On the other hand, the proposed YOLOv7-PDM algorithm model can accurately detect and recognize most of the pen stroke features, demonstrating superior performance in bounding box regression and higher confidence levels compared to the YOLOv7 algorithm model. This further proves that the YOLOv7-PDM algorithm model is the most suitable for detecting calligraphy Kai-style characters' pen stroke features.


FIGURE 7
(Continued).

### 4.4.3 Test of replica calligraphy regular script works

To further confirm the effectiveness of the proposed method, tests were carried out using two genuine copies of regular script characters and their corresponding imitations by the same calligrapher. The testing procedure involved extracting individual characters from the two authentic copies and two imitations separately, following the method mentioned above of creating the dataset. As a result, four sets of character datasets were obtained for detection. Subsequently, the regular script pen-


FIGURE 7
(Continued).

pressure feature detection was performed on each of the four sets of character datasets. Finally, the total number of pen-pressure features for each category of regular script characters in the four datasets was recorded, and the detection results are presented in Table 7.

Table 7 highlights a significant difference in the total stroke feature count of different categories of regular script characters detected using the YOLOv7-PDM algorithm for the authentic and imitation works of Shen in works one and 2. The total stroke feature count for each category in the two authentic works is generally above

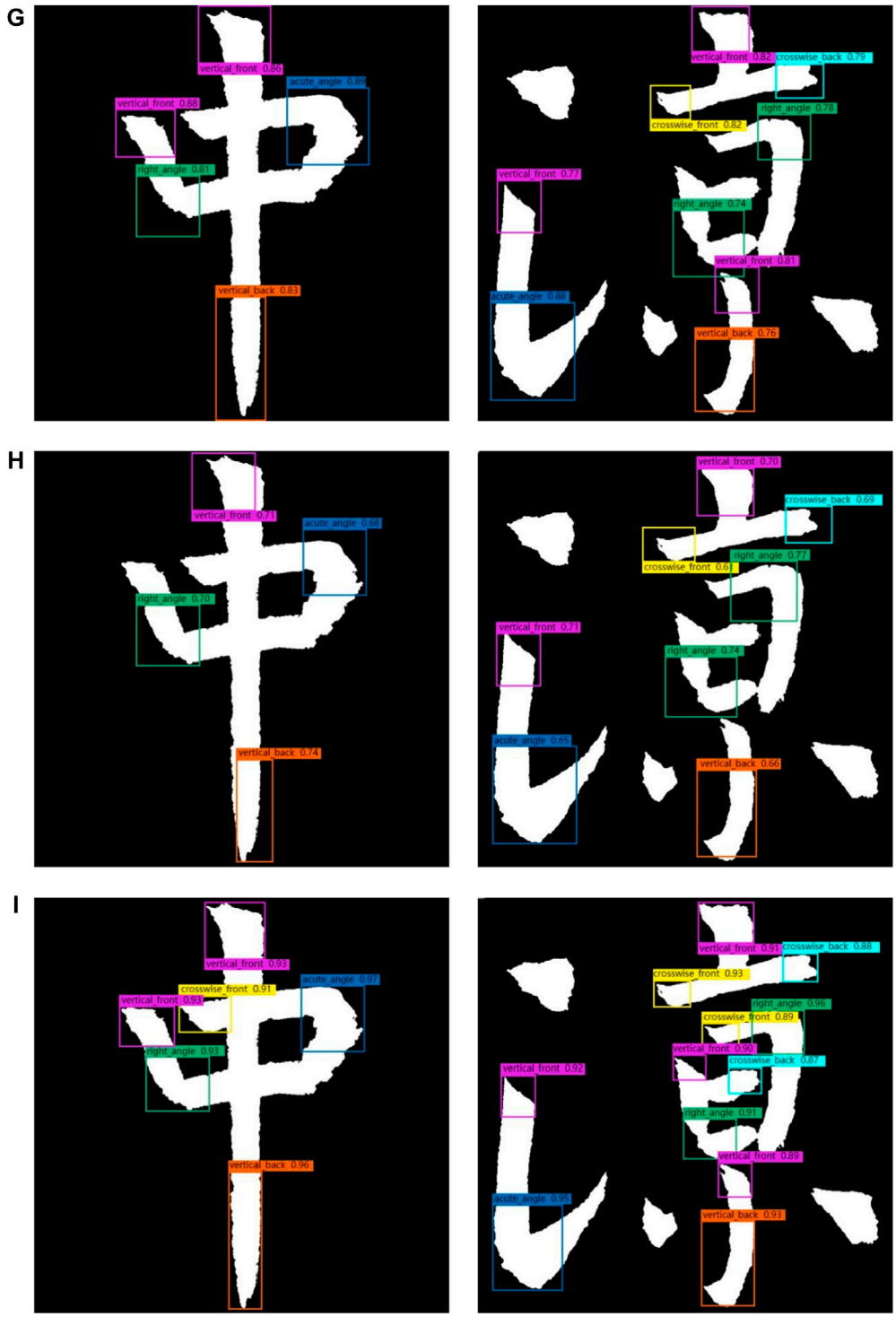



FIGURE 7
(Continued). Comparison of the effects of nine algorithm models: **(A)** YOLOv7; **(B)** YOLOv6; **(C)** YOLOv8; **(D)** DETR; **(E)** Deformable-DETR; **(F)** RT-DETR; **(G)** Faster-RCNN; **(H)** SSD; **(I)** YOLOv7-PDM.

200, while the total for each category in the two imitation works is below 25. This demonstrates that the developed method efficiently differentiates between genuine and counterfeit works of Shen's regular script characters. Moreover, this serves as additional evidence supporting the efficacy of the identification method introduced in this paper.

TABLE 7 Test results of Shen's regular script works identification.

| Calligraphy | Work type | Horizontal start | Horizontal end | Vertical start | Vertical end | Straight angle | Fold angle |
|---|---|---|---|---|---|---|---|
| Works 1 | Genuine | 256 | 354 | 229 | 273 | 215 | 261 |
| | Replica | 11 | 18 | 10 | 13 | 19 | 20 |
| Works 2 | Genuine | 266 | 314 | 329 | 203 | 315 | 191 |
| | Replica | 13 | 19 | 21 | 8 | 17 | 12 |

## 5 Conclusion

This paper presents an enhanced YOLOv7-PDM algorithm model for verifying regular calligraphy script works built upon the YOLOv7 algorithm. Specifically, to avoid the increased complexity of the improved YOLOv7 algorithm, we replaced the convolutional layers in the Backbone part with the PConv module. Reducing the model's parameter count and computational cost (FLOPs) enhanced the algorithm's mAP0.5 and maintained the same single-image detection time. Furthermore, the DyHead dynamic detection head was introduced to enhance the detection accuracy of the YOLOv7 algorithm as much as possible. This improvement increased the algorithm's recognition accuracy without affecting the inference speed. Additionally, to improve the regression capability of the bounding boxes in the YOLOv7 algorithm, we incorporated the MPDioU bounding box loss function. By further improving the overall mAP0.5 value, a recognition accuracy of 94.19% was achieved. By comparing the YOLOv7-PDM algorithm model with eight mainstream algorithms including YOLOv7, YOLOv6, YOLOv8, Deformable-DETR, RT-DETR, Faster-RCNN, SSD, and DETR, we demonstrated that the YOLOv7-PDM algorithm achieved the best performance in terms of mAP0.5 and single-image detection time, accomplishing the improvement goals of the algorithm.

When applying the YOLOv7-PDM algorithm to the authentication of calligraphy regular script works, the genuine works and replicas can be distinguished by comparing the detected feature quantities. Nevertheless, there is scope for enhancing our algorithm as we overlooked special cases like overlapping and intersecting characters in the later stages of calligraphy cursive script works, which directly impacted the accuracy of the model's detection. In upcoming studies, our main goal will be to refine the algorithm and enhance the model's resilience.

## Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

## Author contributions

JC: Conceptualization, Data curation, Investigation, Methodology, Writing–original draft. ZH: Conceptualization, Data curation, Investigation, Methodology, Writing–original draft. XJ: Conceptualization, Data curation, Formal Analysis, Methodology, Validation, Writing–original draft. HY: Funding acquisition, Project administration, Supervision, Validation, Writing–review and editing. WW: Formal Analysis, Funding acquisition, Project administration, Supervision, Writing–review and editing. JW: Funding acquisition, Investigation, Resources, Supervision, Visualization, Writing–review and editing. XW: Formal Analysis, Resources, Software, Validation, Visualization, Writing–review and editing. ZX: Project administration, Resources, Software, Visualization, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded in part by the National Key Research and Development Project of China (grant number: 2018YFA0902900), the Basic Research Program of Guangzhou City of China (grant number 202201011692), and the Guangdong Water Conservancy Science and Technology Innovation Project (grant number 2023-03).

## Acknowledgments

The authors would like to express their thanks to the Guangzhou Institute of Advanced Technology for helping them with the experimental characterization. We want to express our gratitude to the Shenzhen Sanpin Art Museum for providing the calligraphy artwork data.

## Conflict of interest

Author WW was employed by Shenzhen Cas Derui Intelligent Technology Co, Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

# References


1. Yuan BI. Calligraphy. *Calligraphy and Painting Art* (2023) 2023(06):16–29+98.

2. Chen H, Xu B. *Chen Hui and Xu Bangda's calligraphy and painting appraisal research master's thesis*. Qufu: Qufu Normal University (2023).

3. Zhao R. *Research on the authenticity identification of Chinese calligraphy based on style characteristics master's thesis*. Xi'an: Xi'an University of Architecture and Technology (2016).

4. Zhang Y. On the identification of dong qichang's calligraphy——taking the cultural relics in the collection of shandong Museum as an example. *Artwork* (2021) 2021(06):8–15.

5. Lu Q, Liu C. Discussion on the application of handwriting theory to calligraphy forensics. *Leg Syst Soc* (2021) 2021(05):163–5. doi:10.19387/j.cnki.1009-0592.2021.02.166

6. Zeng B. *Research on the authenticity identification method of Chinese calligraphy based on image recognition, master's thesis*. Xi'an: Xi'an University of Architecture and Technology (2015).

7. Chen SC. *Computer-aided authentication of Chinese calligraphy*. Xi'an: Xi'an University of Architecture and Technology (2017).

8. Xiao L. *Evaluation and detection of calligraphy copying characters based on deep learning*. Shanghai: East China University of Science and Technology (2023).

9. Wang H, Huang Y, Cai B. Comparison of image similarity algorithms based on traditional methods and deep learning methods. *Comp Syst Appl* (2024) 33(02):253–64. doi:10.15888/j.cnki.csa.009413

10. Wang W, Jiang X, Yuan H, Chen J, Wang X, Huang Z Research on algorithm for authenticating the authenticity of calligraphy works based on improved EfficientNet network. *Appl Sci* (2024) 14:295. doi:10.3390/app14010295

11. Xu X, Chen B, Wang J. Target detection algorithm for river and lake ships based on improved YOLOv4-Tiny. *Yangtze River* (2023) 54(09):264–71. doi:10.16232/j.cnki.1001-4179.2023.09.035

12. Hu J, Wang C, Yang C. Research on surface hole formation defect detection of aluminum castings based on improved YOLOX algorithm. *Special-cast and Non-ferrous Alloys* (2023) 43(09):1205–9. doi:10.15980/j.tzzz.2023.09.008

13. Mai G, Liang Y, Pan J. Calligraphy font recognition algorithm based on improved DenseNet network. *Comp Syst Appl* (2022) 31(02):253–9. doi:10.15888/j.cnki.csa.008326

14. Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; June 18-22, 2023; Vancouver, Canada (2023). p. 7464–75.

15. Bochkovskiy A, Wang C-Y, Liao H-YM. Yolov4: optimal speed and accuracy of object detection (2020). Available from: https://arxiv.org/abs/2004.10934 (Accessed August 9, 2023).

16. Wang W, Chen J, Huang Z, Yuan H, Peng L, Jiang X, et al. Improved YOLOv7-based algorithm for detecting foreign objects on the roof of a subway vehicle. *Sensors* (2023) 23(23):9440. doi:10.3390/s23239440

17. Chen J, Kao S-hong, He H, Zhuo W, Wen S, Lee C-H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; June 18-22, 2023; Vancouver, Canada (2023). p. 12021–31.

18. Dai X, Chen Y, Xiao B, Chen D, Liu M, Lu Y, et al. Dynamic head: unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; Jun 19-25, 2021; Virtual (2021). p. 7373–82.

19. Siliang M, Yong X. MPDIoU: a loss for efficient and accurate bounding box regression (2023). Available from: https://arxiv.org/abs/2307.07662 (Accessed September 12, 2023).

20. Zhang X A *study on the retrieval of Chinese digital calligraphy and the identification of the authenticity of works*. Hangzhou: Zhejiang University (2006). Master's thesis.

21. Li C, Li L, Jiang H, Weng K, Geng Y, Liang L, et al. YOLOv6: a single-stage object detection framework for industrial applications (2022). Available from: https://arxiv.org/abs/2209.02976 (Accessed August 13, 2023).

22. Oh G, Lim S. One-stage brake light status detection based on YOLOv8. *Sensors* (2023) 23(17):7436. doi:10.3390/s23177436

23. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable detr: Deformable transformers for end-to-end object detection (2020). Available from: https://arxiv.org/abs/2010.04159 (Accessed June 8, 2023).

24. Lv W, Xu S, Zhao Y, Wang G, Wei J, Cui C, et al. Detrs beat yolos on real-time object detection (2023). Available from: https://arxiv.org/abs/2304.08069 (Accessed July 25, 2023).

25. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems 28 IEEE Trans Pattern Anal Mach Intell; Dec 7-12, 2015; Montreal, Quebec, Canada, 39 (2015). p. 1137–49.

26. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. Ssd: single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Proceedings, Part I 14; October 11–14, 2016; Amsterdam, The Netherlands. Springer International Publishing (2016). p. 21–37. Available from: https://arxiv.org/pdf/1512.02325.pdf. (Accessed July 25, 2023)

27. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European conference on computer vision; Aug 23-28, 2020; SEC, Glasgow (2020). p. 213–29. Available from: https://arxiv.org/pdf/2005.12872.pdf (Accessed July 26, 2023).

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 31 May 2024
DOI 10.3389/fphy.2024.1411559



OPEN ACCESS

EDITED BY
Jicheng Wang,
Jiangnan University, China

REVIEWED BY
Zhiqin Zhu,
Chongqing University of Posts and Telecommunications, China
Yu Liu,
Hefei University of Technology, China

*CORRESPONDENCE
Minghong Xie,
minghongxie@163.com

RECEIVED 03 April 2024
ACCEPTED 14 May 2024
PUBLISHED 31 May 2024

CITATION
Zhang S and Xie M (2024), MIPANet: optimizing RGB-D semantic segmentation through multi-modal interaction and pooling attention.
*Front. Phys.* 12:1411559.
doi: 10.3389/fphy.2024.1411559



# MIPANet: optimizing RGB-D semantic segmentation through multi-modal interaction and pooling attention

Shuai Zhang and Minghong Xie*

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China

The semantic segmentation of RGB-D images involves understanding objects appearances and spatial relationships within a scene, which necessitates careful consideration of multiple factors. In indoor scenes, the presence of diverse and disorderly objects, coupled with illumination variations and the influence of adjacent objects, can easily result in misclassifications of pixels, consequently affecting the outcome of semantic segmentation. We propose a Multi-modal Interaction and Pooling Attention Network (MIPANet) in response to these challenges. This network is designed to exploit the interactive synergy between RGB and depth modalities, aiming to enhance the utilization of complementary information and improve segmentation accuracy. Specifically, we incorporate a Multi-modal Interaction Module (MIM) into the deepest layers of the network. This module is engineered to facilitate the fusion of RGB and depth information, allowing for mutual enhancement and correction. Moreover, we introduce a Pooling Attention Module (PAM) at various stages of the encoder to enhance the features extracted by the network. The outputs of the PAMs at different stages are selectively integrated into the decoder through a refinement module to improve semantic segmentation performance. Experimental results demonstrate that MIPANet outperforms existing methods on two indoor scene datasets, NYU-Depth V2 and SUN-RGBD, by optimizing the insufficient information interaction between different modalities in RGB-D semantic segmentation. The source codes are available at https://github.com/2295104718/MIPANet.

KEYWORDS

**RGB-D semantic segmentation, attention mechanism, feature fusion, multi-modal interaction, feature enhancement**

## 1 Introduction

In recent years, Convolutional Neural Networks (CNN) have been widely used in image semantic segmentation, and more and more high-performance models have gradually replaced the traditional semantic segmentation methods. With the introduction of Fully Convolutional Neural Networks (FCN) [1, 2], which has shown great potential in semantic segmentation tasks, many researchers have proposed improved semantic segmentation models based on this way.

The advent of depth sensors and cameras [3] has expanded image research from RGB colour images to RGB-Depth (RGB-D) images, which include depth information. RGB images provide details of object appearance, such as colour and texture, while depth images

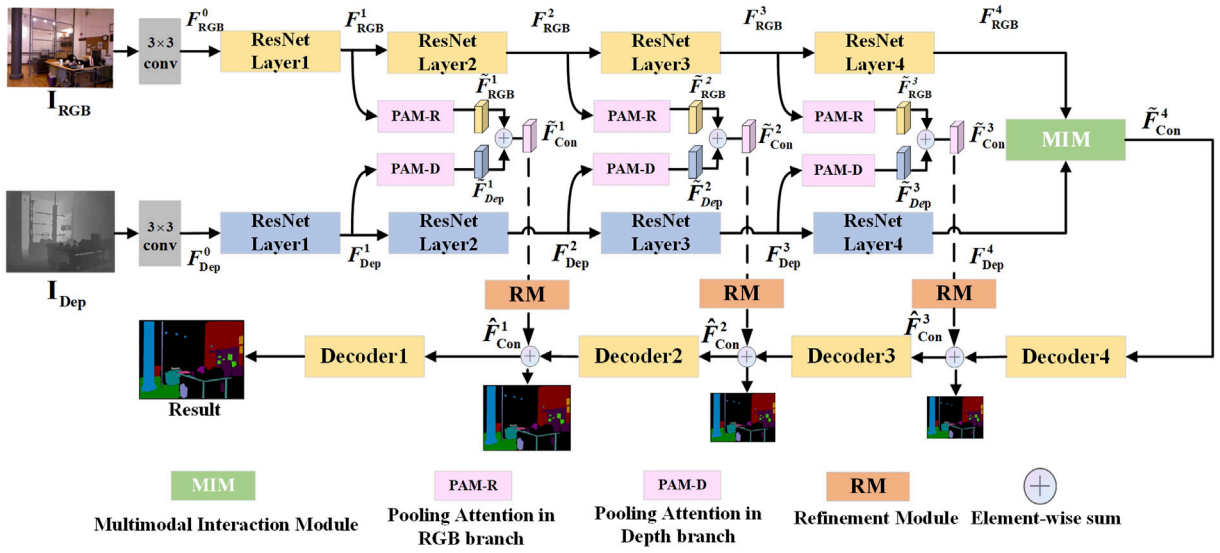



FIGURE 1
Overall architecture of the proposed network is outlined as follows: Each PAM-R or PAM-D across various levels of the network shares an identical configuration but implements distinct operations on two separate branches, yielding RGB and depth features. There are represented as $\tilde{F}^{n}_{RGB}$ and $\tilde{F}^{n}_{Dep}$. After performing an element-wise sum, we obtain $\tilde{F}^{n}_{Con}$, where n indicating the network level. The MIM processes RGB and depth features obtained from the ResNetLayer4 and integrates the fusion result $\tilde{F}^{4}_{Con}$ into the decoder.

contribute essential three-dimensional geometry information absent in RGB images, which is particularly valuable for indoor scene analysis. The fusion of these two modalities of image information would contribute to enhancing the accuracy of indoor scene semantic segmentations. [4, 5] directly concatenated RGB and depth features to create a four-channel input, resulting in improved semantic segmentation accuracy. [6] converted depth images into three channels to an HHA image which consisted of the horizontal disparity, height above ground, and angle of surface normals. Subsequently, RGB features and HHA features are fed into parallel CNNs to predict probability maps for two separate semantic segmentation. These feature maps were then fused in the final layer of the network to produce the ultimate segmentation result. Park et al. [7] and Lee et al.[8] fused the RGB features and depth features through a concatenation process. Eigen et al. [9] and Wang et al. [10] merged the RGB and depth features through directly summation. These methods fail to fully utilize the complementary information between modalities by simply summing or concatenating RGB and depth features. Shu, Li and Bai et al. [11-15] mapped text and image data to a common hash space and facilitated the interaction of information between text and images, which enhanced the performance of cross-modal retrieval. Yang et al. [16] adopted different enhancement mechanisms for RGB data and depth data, including pixel difference convolution techniques, to more effectively handle depth information. Zhao et al. [17] proposed to coordinate attention and cross-modal attention mechanisms, achieving efficient fusion of RGB and depth features and enhancing cross-modal information exchange. Yang et al. [18] developed a dual-stage refinement network (DRNet). In the initial stage, the network focuses on rough localization and feature extraction, while in the advanced stage, it concentrates on feature refinement and precise segmentation. This architecture enables more effective object boundary recovery and definition in complex scenes, thereby improving the accuracy of semantic segmentation. These methods are more effective. However, they use similar or identical operations for extracting RGB and depth features, which does not fully consider the modal differences between RGB and depth images. Moreover, they overlook the interaction between modalities, failing to maximize the complementary nature of the information from different modalities.

To solve the above problems, we propose a Multi-modal Interaction and Pooling Attention Network (MIPANet) for RGB-D semantic segmentation of indoor scenes, as illustrated in Figure 1. The proposed network adopts an encoder-decoder architecture, including two innovative modules: the Multi-modal Interaction Module (MIM) and the Pooling Attention Module (PAM). The encoder is composed of two identical CNN branches, used for extracting RGB features and depth features, respectively. In this study, RGB and depth features are incrementally extracted and fused across various network levels, utilizing spatial disparities and semantic correlations between multimodal features to optimize semantic segmentation results. In the PAM, we employ different feature enhancement strategies for RGB features and depth features. For RGB features, we use global average pooling to make the network focus on the spatial location information of RGB features. For depth features, we employ a two-step pooling operation to replace the global average pooling, aiming to guide the network during depth feature extraction to focus on the most salient parts in each channel. This allows the network to emphasize feature channels containing contours, edge information, and others, thereby enhancing feature representation. Meanwhile, it enables flexible adjustment of the output size and mitigates the impact of large outliers on the results. In the MIM, through cross-modal attention, we enable the RGB features and depth features to learn different information from each other, thereby reducing the disparity between the two modalities and enhancing their

interaction. In the upsampling stage, we design a refinement module (RM) to refine the output of the PAM. This operation enriches the information of the fused features, thereby improving the accuracy of segmentation. The main contributions of this work can be summarized as follows:

(1) We propose an end-to-end multi-modal fusion network, MIPANet, incorporating multi-modal interaction and pooling attention. This method significantly enhances the feature representation of RGB and depth features, effectively focusing on regions with adjacent objects and object overlap regions in the image. Moreover, the proposed method enhances the interaction between RGB and depth features, reduces the feature disparity between modalities, enriches the fused features, and improves semantic segmentation performance.
(2) We design the MIM and PAM. Within the MIM, a cross-modal feature interaction and fusion mechanism is developed. RGB and depth features are collaboratively optimized using attention maps to extract partially detailed features. In addition, the PAM augments the extraction of RGB and depth features through distinct operations, acting as an essential supplement of information in the decoder. It facilitates feature upsampling and restoration via the RM module, ensuring a comprehensive enhancement and integration of critical details for accurate segmentation.
(3) Experimental results confirm the effectiveness of our proposed RGB-D semantic segmentation network in accurately handling indoor images in complex scenarios. The proposed model demonstrates superior semantic segmentation performance compared to other methods on the publicly available NYU-Depth V2 and SUN RGB-D datasets. The visualization results demonstrate that our method focuses more effectively on regions of the image where neighbouring objects may be similar and overlap between objects, resulting in more accurate segmentation outcomes in these regions.

# 2 Related works

## 2.1 RGB-D semantic segmentation

With the widespread application of depth sensors and depth cameras in the field of depth estimation [19-21], people can obtain the depth information of the scene more conveniently, and the research on the image is no longer limited to a single RGB image. The RGB-D semantic segmentation task is to effectively integrate RGB features and depth features to improve segmentation accuracy, especially in some indoor scenes. He et al. [4] proposed an early fusion approach, which simply concatenates an image's RGB and depth channels as a four-channel input to the convolutional neural network. Gupta et al. [6] separately input RGB features and HHA features into two CNNs for prediction and perform fusion in the final stage of the network, and Hazirbas et al. [22] introduced an encoding-decoding network, employing a dual-branch RGB encoder to extract features separately from RGB images and depth images. The studies mentioned above employed equal-weight concatenation or summation operations to fuse RGB and depth features without fully leveraging the complementary information between different modalities. In recent years, some research has proposed more effective strategies for RGB-D feature fusion. Hu et al. [23] utilized a three-branch encoder that includes RGB, Depth, and Fusion branches, efficiently collecting features without breaking the original RGB and deep inference branches. Seichter et al. [24] have presented an efficient RGB-D segmentation approach, characterised by two enhanced ResNet-based encoders utilising an attention-based fusion for incorporating depth information. Fu et al. [25] proposed a joint learning module that learns simultaneously from RGB and depth maps through a shared network, enhancing the model's generalization ability. Fu et al. [25] proposed a joint learning module that learns simultaneously from RGB and depth maps through a shared network, enhancing the model's generalization ability. Zhang et al. [26] proposed a multi-task shared tube structure that aggregates multi-task features into the decoder, improving the learning results for each task. Chen et al. [27] proposed the S-Conv operator, which introduces spatial information to guide the weights of the convolution kernel, thereby adjusting the receptive field, enhancing geometric perception capabilities, and improving segmentation results. Our MIPANet implements a dual-branch convolutional network that performs distinct operations in the middle and final layers of the network to fully utilize the complementary information of different modalities.

## 2.2 Attention mechanism

In recent years, attention [28-34] has been widely used in computer vision and other fields. Vaswani et al. [28] proposed the self-attention mechanism, which has had a profound impact on the design of the deep learning model. Fu et al. [30] proposed DANet, which can adaptively integrate local features and their global dependencies. Wang et al. [35] utilized spatial attention in an image classification model. Through the backpropagation of a convolutional neural network, they adaptively learned spatial attention masks, allowing the model to focus on the significant regions of the image. Hu et al. [36] proposed channel attention, which adaptively learns the importance of each feature channel through a neural network. Woo et al. [33] incorporated two attention modules that concurrently capture channel-wise and spatial relationships. Wang et al. [37] introduced a straightforward and efficient "local" channel attention mechanism to minimize computational overhead. Qiao et al. [38] introduced a multi-frequency domain attention module to capture information across different frequency domains. Similarly, Ding et al. [39] proposed a contrastive attention module designed to amplify local saliency. Building upon this foundation, Huang et al. [40] proposed a cross-attention module that consolidates contextual information both horizontally and vertically, which can gather contextual information from all pixels. These methods have demonstrated significant potential in single-modality feature extraction. To effectively leverage the complementary information between different modalities, this paper introduces a pooling attention module that learns the differential information between two distinct modalities and fully exploits the intermediate-level

features in the convolutional network and the semantic dependencies between modalities.

## 2.3 Cross-modal interaction

With the development of sensor technology, different types of sensors can provide a variety of modal information for semantic segmentation tasks. The information interaction between RGB and other modalities can improve the performance of multimodal tasks [21, 41–48]. Specifically, Li et al. [21, 41, 42], and Xiao et al. [44] improved the quality of infrared and visible image fusion through cross-modal interaction between RGB image and infrared image. Xiang et al. [45] used a single-shot polarization sensor to build the first RGB-P dataset, incorporated polarization sensing to obtain supplementary information, and improved the accuracy of segmentation for many categories, especially those with polarization characteristics, such as glass. Shen et al. [46] proposed a novel pyramid graph network targeting features, which is closely connected behind the backbone network to explore multi-scale spatial structural features. Shen et al. [47] proposed a structure where graphs and transformers interact constantly, enabling close collaboration between global and local features for vehicle re-identification. Zhuang et al. [48] proposed a network consisting of a two-streams (LiDAR stream and camera stream), which extract features from two modes respectively to realize information interaction between RGB and LIDAR modes. In the task of brain tumor image segmentation, Zhu et al. [49] proposed a new architecture that included an improved Swin Transformer semantic segmentation module, an edge detection module, and a feature fusion module. This design effectively merged deep semantic and edge features, leveraging multi-modal information to integrate global spatial data. Furthermore, Zhu et al. [50] introduced the SDV-TUNet, a model that enriched the network's capacity to handle information by utilizing multi-modal MRI data. They also introduced a multi-level edge feature fusion (MEFF) module, emphasizing the importance of edge information at different levels, which significantly enhanced the precision and efficiency of 3D brain tumor segmentation. Liu et al. [51, 52], fused multi-modal magnetic resonance imaging (MRI) using an adversarial learning framework, treating image fusion as an additional regularization method to aid feature learning, effectively integrating multi-modal features to enhance the model's segmentation performance. Therefore, to fully exploit the features of RGB and Depth images, we advocate for information exchange between these two modalities to leverage their complementary information, thereby enhancing the performance of RGB-D semantic segmentation models.

# 3 Methods

## 3.1 Overview

Figure 1 depicts the overall structure of the proposed network. The architecture follows an encoder-decoder design, employing skip connections to facilitate information flow between encoding and decoding layers. The encoder comprises a dual-branch convolutional network, which is used to extract RGB features and depth features. We utilize two pre-trained ResNet50 networks as the backbone, which exclude the final global average pooling layer and fully connected layers. Subsequently, a decoder is employed to upsample the features and integrate them, progressively restoring image resolution.

## 3.2 Network structure

Given a RGB image $I_{RGB} \in \mathbb{R}^{h\times w\times 3}$, and a Depth image $I_{Dep} \in \mathbb{R}^{h\times w\times 1}$, 3 × 3 convolution is used to extract them shallow features $\boldsymbol{F}^0_{RGB}$ and $\boldsymbol{F}^0_{Dep}$, which can be expressed as Eqs 1 and 2:

$$\boldsymbol{F}^0_{RGB} = Conv_{3\times 3}(I_{RGB}), \tag{1}$$

$$\boldsymbol{F}^0_{Dep} = Conv_{3\times 3}\left(I_{Dep}\right), \tag{2}$$

where $Conv_{3\ \times\ 3}$ denotes 3 × 3 convolution.

The network mainly consists of a four-layer encoder-decoder and introduces two designed modules: MIM and PAM. PAM implements different operations on RGB and depth branches, named PAM-R and PAM-D, respectively. PAM-R refers to PAM in the RGB branch, while PAM-D refers to the PAM in the depth branch. Each layer of the encoder is a ResNetLayer. After $\boldsymbol{F}^0_i$ passing through the ResNetLayer, $\boldsymbol{F}^n_i$ is obtained, the $n$th layer of the encoder can be expressed as Eq. 3:

$$\boldsymbol{F}^n_i = H^n_i\left(\boldsymbol{F}^{n-1}_i\right), \tag{3}$$

where $H^n_i$ ($n$ = 1, 2, 3, 4) represents the $n$th ResNetLayer, $i \in \{RGB, Dep\}$ denotes the RGB feature or Depth feature. Specifically, the RGB features and depth features of the first three layers in the ResNet encoder are fed into the PAM. PAM enhances features by performing different operations on RGB features and depth features, resulting in $\tilde{\boldsymbol{F}}^n_{RGB}$ and $\tilde{\boldsymbol{F}}^n_{Dep}$, where $n$ = 1, 2, 3. Subsequently, the two features are combined by element-wise addition to obtain $\tilde{\boldsymbol{F}}^n_{Con}$, containing rich spatial location information. Furthermore, the final RGB and depth features from the ResNetLayer4 encoder are fed into the MIM to capture complementary information within these two modalities. The output features of the MIM are then fed into the decoder, where each upsampling layer consists of two 3 × 3 convolutional layers. These layers are followed by batch normalization (BN) and ReLU activation, with each upsampling layer doubling the feature spatial dimensions while halving the number of channels.

## 3.3 Pooling attention module

Within the low-level features extracted by the convolutional neural network, we capture the fundamental attributes of the input image. These low-level features are critical in modelling the image's foundational characteristics. However, they lack semantic information from the deep-level neural network, such as object shapes and categories. At the same time, during the upsampling process in the decoding layer, there is a risk of losing certain semantic information as the image resolution increases. To address this issue, we introduce the Pooling Attention Module

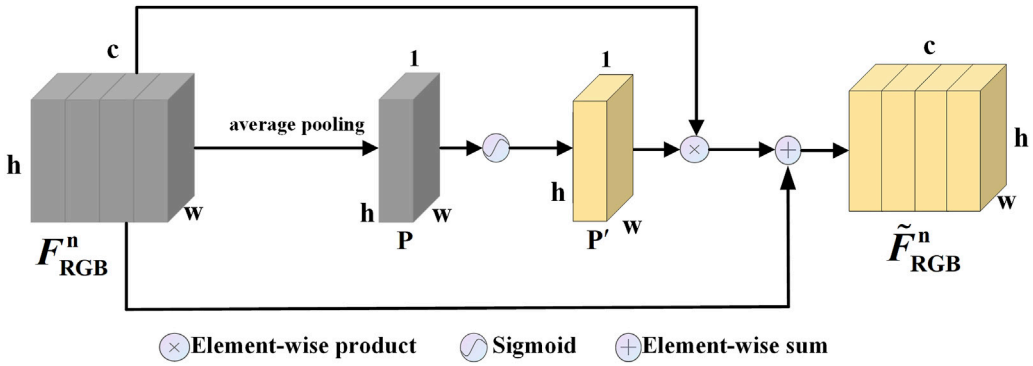



FIGURE 2
Structure of PAM in the RGB branch, referred to as PAM-R. Given an input feature $F^n_{RGB}$, it is first processed through an average pooling operation to obtain $P$. Subsequently, $P$ undergoes a sigmoid activation to produce $P'$. The activated feature $P'$ is then element-wise product with the original input feature $F^n_{RGB}$ to yield a preliminary result, which is further added to the initial feature $F^n_{RGB}$ to generate the final output $\tilde{F}^n_{RGB}$.

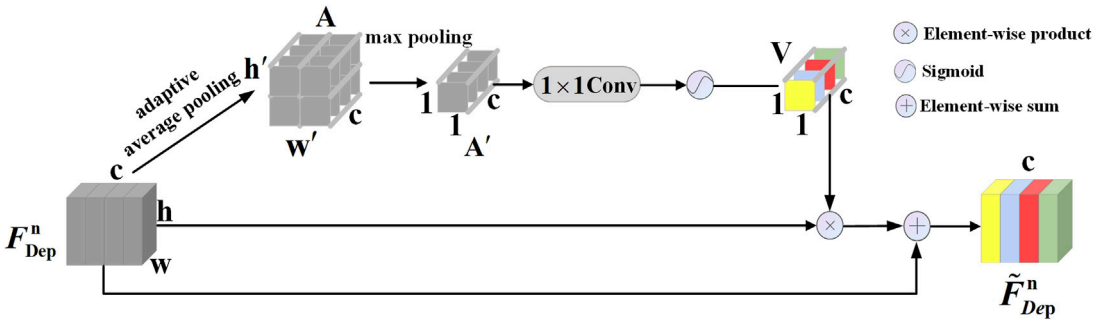



FIGURE 3
Structure of PAM in the depth branch, referred to as PAM-D. The input feature $F^n_{Dep}$ first passes through an adaptive pooling operation, resulting in $A$. This is followed by a max pooling operation to produce $A'$. The output $A'$ then goes through a 1 × 1 convolution and a sigmoid activation to yield the weight vector $V$ (e.g., yellow) between 0 and 1. This $V$ is element-wise product with the original feature $F^n_{Dep}$, and the product is subsequently added to $F^n_{Dep}$ to produce the final output $\tilde{F}^n_{Dep}$.

(PAM). For RGB features, we utilize average pooling to average the information across all channels at each spatial location. This method highlights the importance of each position, aiding in the better capture of key spatial features such as edges and textures. For depth features, we opt for max pooling, which accentuates the most significant signals within each channel. This effectively enhances the model's response to crucial depth information while suppressing less important channels. This approach allows us to more precisely identify and emphasize important features in the depth map, thus improving the overall segmentation accuracy. In the decoding layer, the output from the PAM is first processed by the Refinement Module (RM), effectively compensating for information loss during the upsampling process, and increasing the network's attention to specific areas. This strategy improves the accuracy of segmentation results and efficiently maintains the integrity of semantic information. The structure of the PAM in RGB and depth branches are shown in Figures 2, 3, respectively.

The input feature $F^n_{RGB} \in \mathbb{R}^{h\times w\times c}$ denotes the RGB feature passes through average pooling to reduce the number of channels in the feature map, which can be expressed as Eq. 4:

$$P = H_{avg}\left(F^n_{RGB}\right), \tag{4}$$

where $P \in \mathbb{R}^{h\times w\times 1}$ represents the feature map that has aggregated the information across all channels at each position. $H_{avg}$ denotes the global average pooling operation. $h$, $w$ represent the height and width of the feature map. Then we get the weight vector $P' \in \mathbb{R}^{h\times w\times 1}$ by sigmoid activation, which can be expressed as Eq. 5:

$$P' = Sigmoid\left(P\right), \tag{5}$$

Then, we perform an Element-wise product for $F^n_{RGB}$ and $P'$, and the result $\tilde{F}^n_{RGB}$ can be expressed as Eq. 6:

$$\tilde{F}^n_{RGB} = F^n_{RGB} + \left(F^n_{RGB} \otimes P'\right), \tag{6}$$

where ⊗ denotes the Element-wise product. Through the PAM in the RGB branch, the original feature map, after being weighted by spatial attention, is enhanced at important spatial locations, while less important locations are relatively suppressed, thus enabling the network to focus more on spatial regions that are useful for semantic segmentation.

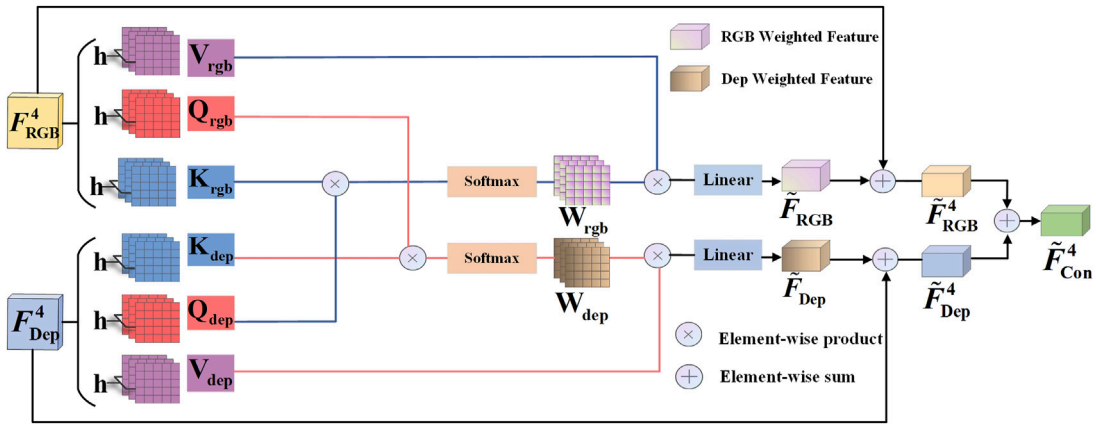



FIGURE 4
Structure of the MIM. The RGB feature and the depth feature undergo linear transformations to generate two sets of Q, K, V (e.g., blue line) for multi-head attention, where h denotes the number of attention heads set to 8. The weighted summation of input features $F^4_{RGB}$ and $F^4_{Dep}$ yields $\tilde{F}^4_{RGB}$ and $\tilde{F}^4_{Dep}$, which are then element-wise added to obtain the output result $\tilde{F}^4_{Con}$.

The input feature $F^n_{Dep} \in \mathbb{R}^{h\times w\times c}$ denotes the Depth feature passes through adaptive average pooling to reduce the feature map to a smaller dimension, which can be expressed as Eq. 7:

$$A = H_{ada}\left(F^n_{Dep}\right), \tag{7}$$

where $A \in \mathbb{R}^{h'\times w'\times c}$ represents the feature map that has been resized by adaptive averaging pooling, $H_{ada}$ denotes the adaptive average pooling operation. $h'$, $w'$ represent the height and width of the output feature map, which we set $h' = 2$ and $w' = 2$. Then, we get the output features $A'$ by max pooling the features after dimensionality reduction, which can be expressed as Eq. 8:

$$A' = H_{max}(A), \tag{8}$$

where $A' \in \mathbb{R}^{1\times1\times c}$ represents the pooling result and then $A'$ undergoes a 1 × 1 convolution and then activation with the sigmoid function, getting a weight vector $V \in \mathbb{R}^{1\times1\times c}$ value between 0 and 1. $H_{\max}$ denotes the max pooling operation. Finally, we perform an Element-wise product for $F^n_{Dep}$ and $V$, and the result $\tilde{F}^n_{Dep}$ can be expressed as Eqs 9, 10:

$$V = Sigmoid\left(\Phi(A')\right), \tag{9}$$

$$\tilde{F}^n_{Dep} = F^n_{Dep} + \left(F^n_{Dep} \otimes V\right), \tag{10}$$

where ⊗ denotes the Element-wise product, $\Phi$ denotes 1 × 1 convolution. The PAM in the depth branch makes the network pay more attention to local regions in the image, such as objects near the background in the scene. Meanwhile, adaptive average pooling can enhance the module's flexibility, accommodating diverse input feature map dimensions and fully retaining spatial position information in depth features. $\tilde{F}^n_{Con}$ in Figure 1 can be expressed as Eq. 11:

$$\tilde{F}^n_{Con} = \tilde{F}^n_{RGB} + \tilde{F}^n_{Dep}, \tag{11}$$

During the upsampling process, $\tilde{F}^n_{Con}$ $(n = 1, 2, 3)$ is fed into the decoder.

## 3.4 Multi-modal interaction module

When adjacent objects in an image share similar appearances, distinguishing their categories becomes challenging. Factors such as lighting variations and object overlap, especially in the corners, can lead to their blending with the background. This complexity makes it difficult to precisely identify object edges, leading to misclassification of the object as part of the background. Depth information remains unaffected by lighting conditions and can accurately differentiate between objects and the background based on depth values. Therefore, we design the MIM to supplement RGB information with Depth features. Meanwhile, it utilizes RGB features to strengthen the correlation between RGB and depth features. Depth features excel in capturing object contours and edge information, compensating for the spatial depth information that RGB features lack. Conversely, RGB features play a crucial role in compensating for the deficiencies in depth features, particularly in aspects such as color and texture, thereby enriching the information content of depth features.

MIM achieves dual-mode feature fusion, as depicted in Figure 4. Here, $F^4_{RGB} \in \mathbb{R}^{h\times w\times c}$ and $F^4_{Dep} \in \mathbb{R}^{h\times w\times c}$ correspond to the RGB feature and depth feature from the ResNetLayer4. The feature channels are denoted as "c", and their spatial dimensions are $h \times w$. First, the two feature maps are linearly mapped to generate multi-head query(Q), key(K), and value(V) vectors. Here, "rgb" and "dep" represent the RGB and depth features. These linear mappings are accomplished via fully connected layers, where each attention head possesses its unique weight matrix. For each attention head, we calculate the dot product between two sets of $Q$ and $K$ and then normalize the results to a range between 0 and 1 using the softmax function to get the attention maps $W_{rgb}$ and $W_{dep}$, which can be expressed as Eqs 12, 13:

$$W_{rgb} = Softmax\left(\frac{Q_{rgb}K^T_{dep}}{\sqrt{d}_k}\right) \tag{12}$$

$$W_{dep} = Softmax\left(\frac{Q_{dep}K^T_{rgb}}{\sqrt{d}_k}\right) \tag{13}$$

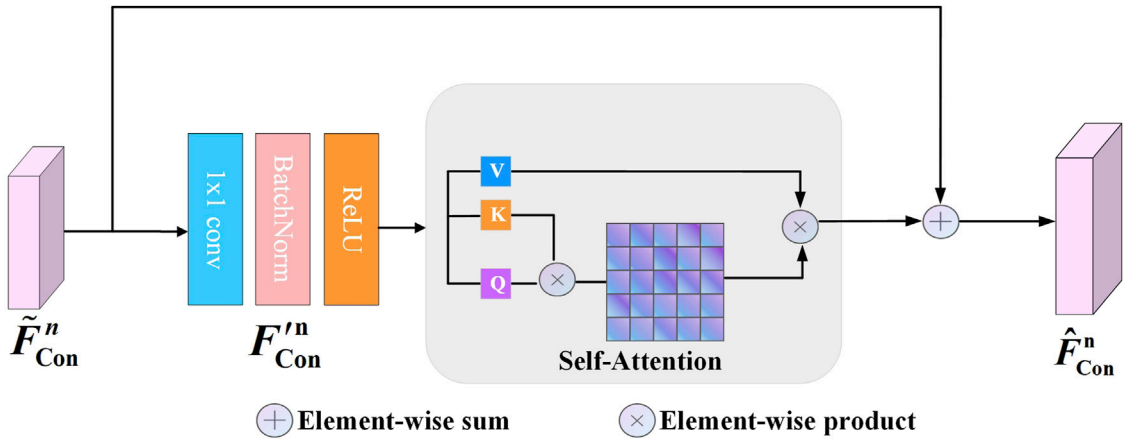



FIGURE 5
Design of the Refinement Module: following a 1 × 1 convolution, BatchNorm, and ReLU activation function, the feature map $F'^{n}_{Con}$ is produced. Then, utilizing a self-attention operation, the weighted feature $\hat{F}^{n}_{Con}$ is derived, enhancing the module's ability to focus on salient aspects of the input.

where $d_k$ represents the dimensionality of the $K$ vector. Then, we calculate the RGB weighted feature $\tilde{F}_{RGB}$ and the depth weighted feature $\tilde{F}_{Dep}$, and the final output features $\tilde{F}^{4}_{RGB}$ and $\tilde{F}^{4}_{Dep}$ are obtained through a residual connection, which can be expressed as Eqs 14, 15:

$$\tilde{F}_{RGB} = W_{rgb} \otimes V_{rgb} \tag{14}$$

$$\tilde{F}^{4}_{RGB} = \tilde{F}_{RGB} + F^{4}_{RGB} \tag{15}$$

where $\tilde{F}_{RGB}$ represents the RGB weighted feature, $V_{rgb}$ represents the value vector from the RGB feature, multiplying with weight matrix $W_{rgb}$. $\tilde{F}^{4}_{RGB}$ represents the RGB feature after the fusion with depth feature. Likewise, we get the Eqs 16, 17:

$$\tilde{F}_{Dep} = W_{dep} \otimes V_{dep} \tag{16}$$

$$\tilde{F}^{4}_{Dep} = \tilde{F}_{Dep} + F^{4}_{Dep} \tag{17}$$

where $\tilde{F}_{Dep}$ represents the depth weighted feature, $V_{dep}$ represents the value vector from the Depth feature, multiplying with weight matrix $W_{dep}$. $\tilde{F}^{4}_{Dep}$ represents the depth feature after the fusion with RGB feature, ⊗ represents the Element-wise product. Finally, we can obtain the MIM output through Element-wise sum, which can be expressed as Eq. 18:

$$\tilde{F}^{4}_{Con} = \tilde{F}^{4}_{RGB} + \tilde{F}^{4}_{Dep} \tag{18}$$

## 3.5 Refinement module

RGB features provide rich colour and texture information, while depth features provide spatial and shape information. The fusion of these two types of features can help the network to understand the scene more comprehensively. However, due to the differences between the two modalities, simple addition might introduce some noise, affecting the segmentation results. To address this issue, we propose a Refinement Module (RM) that, through a CBR structure (Convolution; Batch Normalization; ReLU), allows the network to adaptively re-extract and optimize the fused features, filtering out unnecessary information and retaining features that are more useful for semantic segmentation. Moreover, by utilizing self-attention, the global information of the features is enhanced, enabling a better understanding of the global structure of the input features, thereby improving performance. The structure of the RM is shown in Figure 5.

As shown in Figure 5, $\tilde{F}^{n}_{Con}$ is processed by the CBR operation to generate $F'^{n}_{Con}$, which can be expressed as Eq. 19:

$$F'^{n}_{Con} = CBR\left(\tilde{F}^{n}_{Con}\right) \tag{19}$$

where $n$ = 1, 2, 3. $CBR$ represents a 1 × 1 convolution followed by Batch Normalization and ReLU activate function. Then, $F'^{n}_{Con}$ is linearly mapped to generate query(Q), key(K), and value(V) vectors. Through a self-attention module, the final output result is generated, which can be expressed as Eq. 20:

$$\hat{F}^{n}_{Con} = Softmax\left(\frac{QK^{T}}{\sqrt{d}_{k}}\right) \otimes V + \tilde{F}^{n}_{Con} \tag{20}$$

RM further extracts and refines the fused features to enhance the feature representation, and $\hat{F}^{n}_{Con}$ is fed into the decoder.

## 3.6 Loss function

In this paper, the network performs supervised learning on four different levels of decoding features. We employ nearest-neighbor interpolation to reduce the resolution of semantic labels. Additionally, 1 × 1 convolutions and Softmax functions are utilized to compute the classification probability for each pixel within the output features from the four upsample layers, respectively. The loss function $\mathcal{L}_i$ of layer i is the pixel-level cross entropy loss, which can be expressed as Eq. 21:

$$\mathcal{L}_i = -\frac{1}{N_i} \sum_{\forall p,q} Y(p,q) \log\left(Y'(p,q)\right) \tag{21}$$

where $N_i$ denotes the number of pixels in layer $i$. $p$, $q$ represent the coordinate positions of each pixel in the image. Specifically, p refers to the row coordinate of the pixel, while q refers to the column coordinate. $Y'$ is the classification probability of the output, and $Y$ is the label category. The final loss function $\mathcal{L}$ of the network is obtained by summing the pixel-level loss functions of the four decoding layers, which can be expressed as Eq. 22:

$$\mathcal{L} = \sum_{i=1}^{4} \mathcal{L}_i \tag{22}$$

By optimizing the above loss function, the network can get the final segmentation result.

# 4 Experimental results and analysis

## 4.1 Experimental setup

NYU-Depth V2 dataset [53] and SUN RGB-D dataset [54] are used to evaluate the proposed method. NYU-Depth V2 dataset is a widely used indoor scene understanding dataset for computer vision and deep learning research. It is an aggregation of video sequences from various indoor scenes recorded by RGB-D cameras from the Microsoft Kinect and is an updated version of the NYU-Depth dataset published by Nathan Silberman and Rob Fergus in 2011. It contains 1,449 RGB-D images, depth images, and semantic tags in the indoor environment. The dataset includes different indoor scenes, scene types, and unlabeled frames, and each object can be represented by a class and an instance number. SUN RGB-D dataset contains image samples from multiple scenes, covering various indoor scenes such as offices, bedrooms, and living rooms. It has 37 categories and contains 10,335 RGB-D images with pixel-level annotations, of which 5,285 are used as training images and 5,050 are used as test images. This special dataset is captured by four different sensors: Intel RealSence, Asus Xtion, Kinect v1, and v2. Besides, this densely annotated dataset includes 146,617 2D polygons, 64,595 3D bounding boxes with accurate object orientations, and a 3D room layout as well as an imaged-based scene category.

We evaluate the results using two standard metrics, Pixel Accuracy (Pix. Acc) and Mean Intersection Over Union (mIoU). Pix. Acc refers to pixel accuracy, which is the simplest metric that represents the proportion of correctly labelled pixels to the total number of pixels, which can be expressed as Eq. 23:

$$Pix.Acc = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}}. \tag{23}$$

where $p_{ii}$ means to predict the correct value, and $p_{ij}$ means to predict $i$ to $j$. $k$ represents the number of categories. In addition, Intersection over Union (IoU) is a measure of semantic segmentation, where the IoU ratio of a class is the ratio of the IoU of its true labels and predicted values, while mIoU is the average IoU ratio of each class in the dataset, which can be expressed as Eq. 24:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}. \tag{24}$$

TABLE 1 MIPANet compared to the state-of-the-art methods on the NYU-Depth V2 dataset.

| Method | Backbone | mIoU (%) | Pix.Acc (%) |
|---|---|---|---|
| ESANet | ResNet18 | 48.2 | — |
| IEMNet | Res34NBt1D | 51.3 | 76.8 |
| SGACNet | 2 × Res34NBt1D | 49.4 | 75.6 |
| Z-ACN | ResNet50 | 50.0 | — |
| DynMM | ResNet50 | 51.0 | — |
| RDFNet | 2 × ResNet50 | 47.7 | 74.8 |
| RAFNet | 2 × ResNet50 | 47.5 | 73.8 |
| SA-Gate | 2 × ResNet50 | 50.4 | — |
| ESANet | 2 × ResNet50 | 50.5 | — |
| RedNet | 2 × ResNet50 | 47.2 | — |
| ACNet | 3 × ResNet50 | 48.3 | — |
| SGNet | ResNet101 | 49.6 | 75.6 |
| RDFNet | 2 × ResNet101 | 49.1 | 75.6 |
| ShapeConv | ResNet101 | 51.3 | 76.4 |
| Baseline | 2 × ResNet50 | 47.4 | 75.1 |
| Ours (MIPANet) | 2 × ResNet50 | **52.3** | **77.6** |

The bold values mean the highest results.

TABLE 2 MIPANet compared to the state-of-the-art methods on the SUN RGB-D dataset.

| Method | Backbone | mIoU (%) | Pix.Acc (%) |
|---|---|---|---|
| IEMNet | Res34NBt1D | 48.3 | 81.9 |
| EMSANet | 2 × Res34NBt1D | 48.5 | — |
| RAFNet | 2 × ResNet50 | 47.2 | 81.3 |
| ESANet | 2 × ResNet50 | 48.3 | — |
| RedNet | 2 × ResNet50 | 47.8 | 81.3 |
| ACNet | 3 × ResNet50 | 48.1 | — |
| SGNet | ResNet101 | 47.1 | 81.0 |
| CANet | ResNet101 | 48.3 | 82.0 |
| RDFNet | ResNet101 | 48.2 | 82.3 |
| ShapeConv | ResNet101 | 48.6 | 82.2 |
| RDFNet | 2 × ResNet152 | 47.7 | 81.5 |
| Baseline | 2 × ResNet50 | 45.5 | 81.1 |
| Ours (MIPANet) | 2 × ResNet50 | **49.1** | **82.5** |

The bold values mean the highest results.

where $p_{ij}$ represents the predict $i$ as $j$, and $p_{ji}$ represents the predict $j$ as $i$, $p_{ii}$ means to predict the correct value, $k$ represents the number of categories.

We implement and train our proposed network using the PyTorch framework. To enhance the diversity of the training data, we apply random scaling and mirroring. Subsequently, all

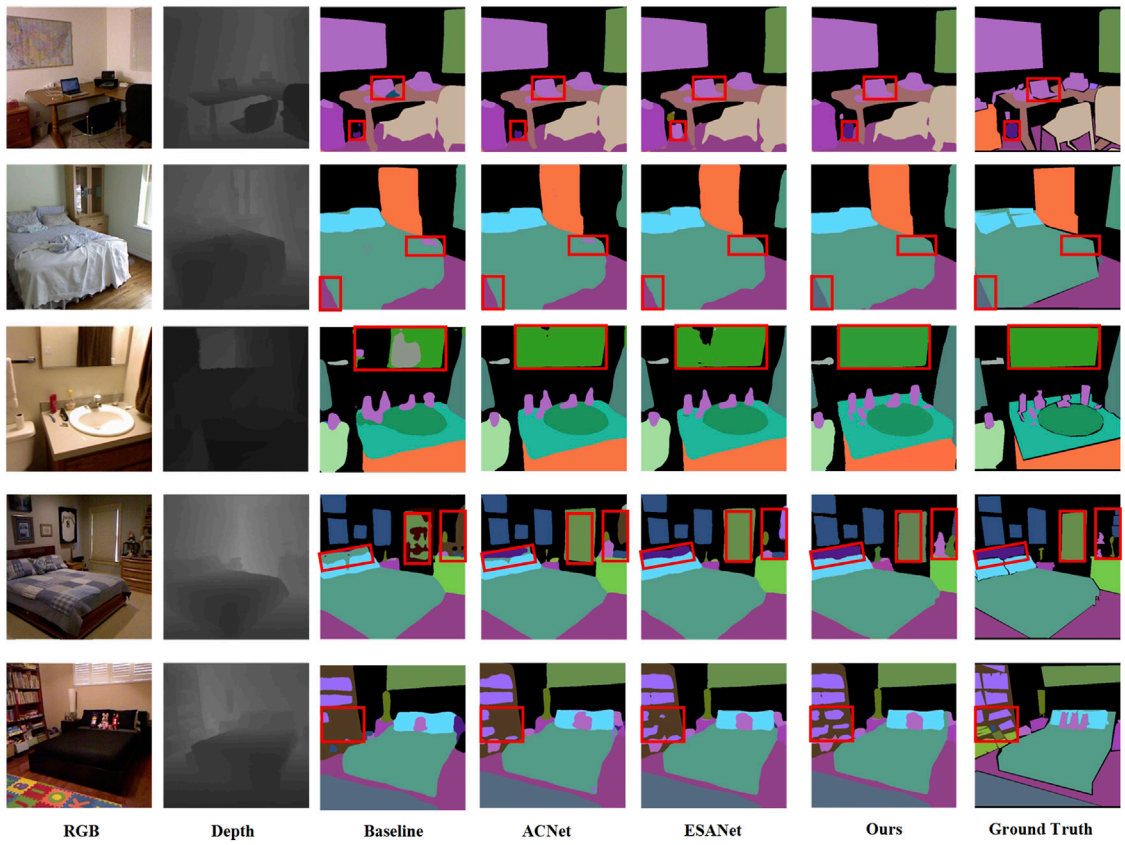



FIGURE 6
Visual comparisons on the NYU-Depth V2 dataset.

RGB and depth images are resized to 480 × 480 for network inputs, and semantic labels are adjusted to sizes of 480 × 480, 240 × 240, 120 × 120, and 60 × 60 for deep supervision training. As the backbone for our encoder, we utilize the ResNet50 pre-trained [55] on the ImageNet dataset [56]. Our baseline model uses two branches as encoders to extract RGB and depth features, respectively, while excluding the PAM during the extraction process. Each branch is composed of four ResNet50 layers. In the final layer of the network, RGB and depth features are merged by element-wise addition, without employing the MIM. The output of element-wise addition is then used as input to the encoder for upsampling operations, resulting in the final segmentation result. To refine the network structure, following [57-59], we adjust it by replacing the 7 × 7 convolution in the input stem with three consecutive 3 × 3 convolutions. The training is conducted on an NVIDIA GeForce GTX 3090 GPU using stochastic gradient descent optimization. Parameters are set with a batch size of 6, an initial learning rate of 0.003, 500 epochs, and momentum and weight decay values of 0.9 and 0.0005, respectively.

## 4.2 Quantitative experimental results on NYU-Depth V2 and SUN RGB-D datasets

To validate the effectiveness of the proposed model in this paper, we compare the proposed method with state-of-the-arts methods (ESANet [24], IEMNet [60], SGACNet [61], Z-ACN [62], DynMM [63], RDFNet [7], RAFNet [64], SA-Gate [65], RedNet [8], ACNet [23], SGNet [27], ShapeConv [66]) on the NYU-Depth V2 dataset. For a fair comparison, we compare our method with others using the ResNet architecture, which employ ResNet with varying depths and quantities.

Table 1 illustrates our superior performance regarding mIoU and Acc metrics compared to other methods. Specifically, with ResNet50 serving as the encoder in our network, the Pix. Acc and mIoU for semantic segmentation on the NYU-Depth V2 test set reached 77.6% and 52.3%. For example, our method improved the mIoU by 4.9% compared to the baseline method. Compared to the runner-up method DynMMXue and Marculescu (2023), which also employs ResNet50, our method achieved a 1.3% improvement. Similarly, compared to the suboptimal method ShapeConvCao et al. (2021), which uses the deeper ResNet101, our method achieved a 1.0% improvement. Our method achieves better results on networks with ResNet50 as the backbone than some methods with ResNet101 as the backbone, showcasing the effectiveness of our carefully designed network structure.

Then, we compare the proposed method with state-of-the-arts methods (IEMNet [60], EMSANet [67], RAFNet [64], ESANet [24], RedNet [8], ACNet [23], SGNet [27], CANet [68], RDFNet [7], ShapeConv [66]) on the SUN RGB-D dataset. As depicted in Table 2, our approach consistently achieves a higher mIoU score on the SUN RGB-D dataset

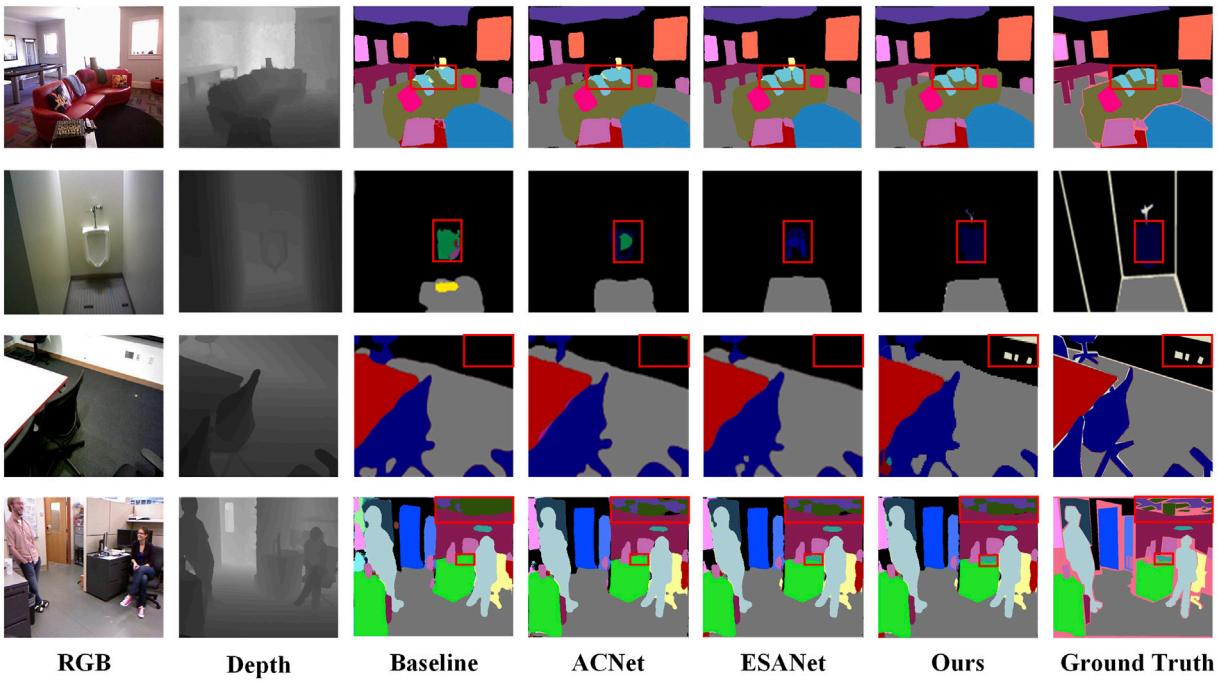



FIGURE 7
Visual comparisons on the SUN RGB-D dataset.

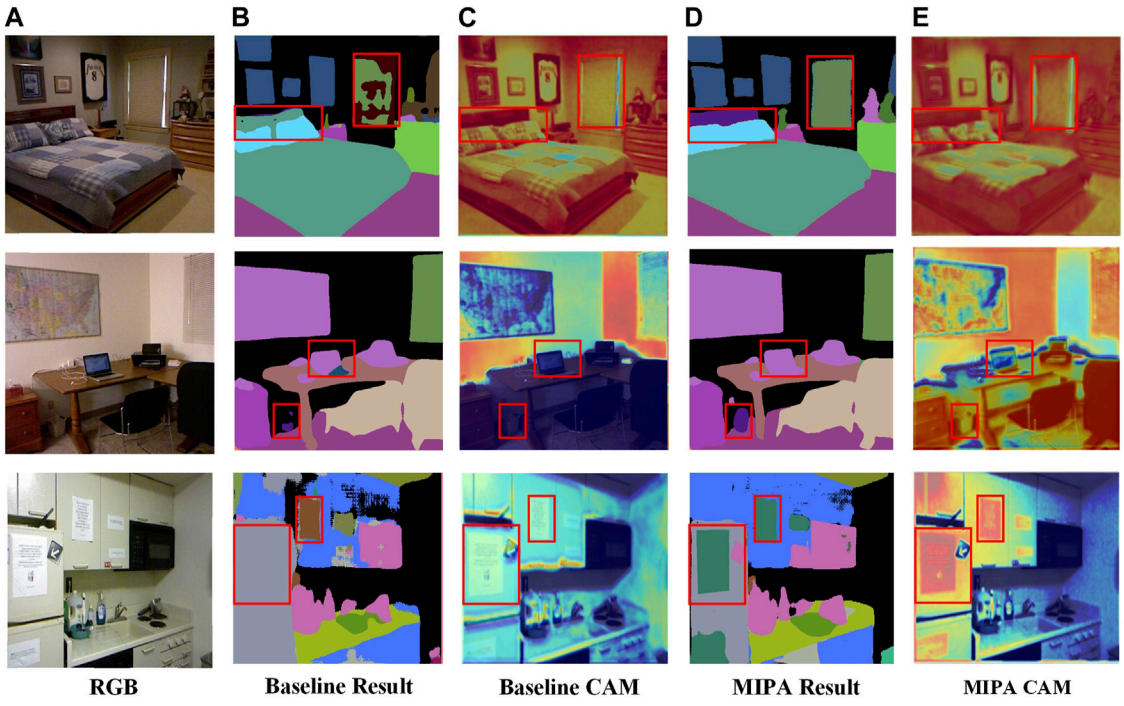



FIGURE 8
Images from left to right represent **(A)** the RGB image, **(B)** the segmentation result of Baseline, **(C)** CAM of Baseline, **(D)** the segmentation results of MIPANet (Ours) and **(E)** CAM of MIPANet. The red box indicates the prominent areas of effect.

TABLE 3 Ablation studies on NYU-Depth V2 dataset for PAM, MIM and RM.

| Method | mIoU (%) | Pix.Acc (%) |
|---|---|---|
| ResNet50 (Baseline) | 47.4 | 75.1 |
| ResNet50 + PAM | 48.9 | 76.0 |
| ResNet50 + PAM + RM | 49.5 | 76.0 |
| ResNet50 + MIM | 51.1 | 77.0 |
| ResNet50 + PAM + MIM | 51.9 | 77.2 |
| ResNet50 + PAM + MIM + RM (Ours) | 52.3 | 77.6 |

TABLE 4 Ablation studies on SUN RGB-D dataset for PAM, MIM and RM.

| Method | mIoU (%) | Pix.Acc (%) |
|---|---|---|
| ResNet50 (Baseline) | 45.5 | 81.1 |
| ResNet50 + PAM | 47.9 | 81.3 |
| ResNet50 + PAM + RM | 48.1 | 81.3 |
| ResNet50 + MIM | 48.3 | 81.5 |
| ResNet50 + PAM + MIM | 48.8 | 82.3 |
| ResNet50 + PAM + MIM + RM (Ours) | 49.1 | 82.5 |

TABLE 5 Performance comparison of the different methods on the number of model parameters, FLOPs and testing time.

| Models | Parameter(M) | FLOPs(G) | Time (ms) |
|---|---|---|---|
| ACNet | 116.6 | 126.3 | 45.0 |
| RedNet | 82.0 | 101.8 | 34.7 |
| RDFNet | 443.8 | 648.7 | 71.9 |
| SA-Gate | 110.6 | 176.5 | 53.1 |
| MIPANet | 360.0 | 634.2 | 62.4 |

than all other methods. For example, our method improved the mIoU by 3.6% compared to the baseline method. Compared to the suboptimal method ESANet [24], which also employs ResNet50, our method achieved a 0.8% improvement. Similarly, compared to the suboptimal method ShapeConv [66], which uses the deeper ResNet101, our method achieved a 0.5% improvement. This observation underscores our module's ability to maintain superior segmentation accuracy, even when dealing with the extensive SUN RGB-D dataset.

## 4.3 Visualization results on NYU-Depth V2 and SUN RGB-D datasets

To visually highlight the advancements made by our method, we provide visualization results of the network on the NYU-Depth V2 dataset and SUN RGB-D datasets, as shown in Figures 6, 7. From left to right, the RGB image, the Depth image, the baseline model results with ResNet50 backbone, ACNet, ESANet, MIPANet (Ours), and Ground Truth.

As shown in Figure 6, compared to the baseline, our method significantly improve segmentation results. Notably, the dashed box in the figure showcases our network enrich with depth information accurately distinguishes objects from the background. For instance, in the visualization results of the fourth image, the baseline erroneously categorizes the mirror on the wall as part of the background, in the visualization results of the second image, the ACNet and the ESANet mistook the carpet for a part of the floor. In contrast, leveraging depth information, our network discerns the distinct distance information of the mirror from the background, leading to a correct classification of the mirror. The proposed method has achieved precise segmentation outcomes in diverse and intricate indoor scenes. Moreover, it excels in segmenting challenging objects like "carpets" and "books" while delivering finer-edge segmentation results.

As shown in Figure 7, our method also achieve better experimental results on the SUN RGB-D dataset. For example, in the second row of Figure 7, the toilet and wall share a similar white color and partially overlap in position, making it difficult for the network to distinguish between them accurately. Compared to other methods, our MIPA approach demonstrates superior effectiveness in segmenting toilet. In the third row of Figure 7, our method accurately segments the power switch on the wall, further demonstrating its effectiveness.

Furthermore, we verify the effectiveness of our method by providing visualization results of class activation mapping (CAM). These visualizations demonstrate that MIPANet effectively focuses on regions containing adjacent or overlapping objects. As shown in Figure 8, compared to the baseline cam Figure 8C, the more prominent red areas in image Figure 8E indicate that our method focuses more on specific regions. For example, in the first row, the adjacent pillow and headboard are highlighted. In the second row, the trash can overlaps with the wall and has a similar color, the computer is close to the tabletop. In the third row, the paper is attached to the refrigerator and cabinet. The network's attention to these areas increased, compared to the baseline segmentation result in Figure 8B, our method achieves more accurate segmentation results, as shown in Figure 8D. The visualization results indicate that our method better focuses on adjacent and overlapping objects in the image.

## 4.4 Ablation study

To investigate the impact of different modules on segmentation performance, we conduct ablation experiments on NYU-Depth V2 and SUN-RGBD datasets, as depicted in Tables 3, 4. For instance, in NYU-Depth V2, our PAM module exhibit a superiority of 1.5% and 0.9% over the baseline concerning mIoU and Pix. Acc indicators. Similarly, our MIM module demonstrate a superiority of 3.7% and 1.9% over the baseline regarding mIoU and Pix. Acc. Additionally, the inclusion of the RM has further improved the performance of the module. The result suggests that each proposed module can independently enhance segmentation accuracy. Our module surpasses the baseline in fusing cross-modal features, yielding superior results on both datasets. Using PAM, MIM and RM modules, we achieve the highest mIoU of 52.3% on the NYU-Depth V2 dataset and the highest mIoU of 49.1% on the SUN RGB-D dataset. The result highlights that our designed modules can be collectively optimized to enhance segmentation accuracy.

### 4.5 Computational complexity analysis

In this section, we analyze the computational complexity of the different methods from three aspects: the number of model parameters, FLOPs, the time required for testing. The results are listed in Table 5. For the evaluation of computational complexity, the size of the input images is standardized to 480 × 640 pixels. The test time is the time taken to process one pair of RGB and depth images. As shown in Table 5, the parameter quantity and FLOPs of our model are moderate. However, compared to the comparison methods, our approach achieves the highest mIoU and exhibits the most visually appealing results.

## 5 Conclusion

In this paper, we tackle a fundamental challenge in RGB-D semantic segmentation—efficiently fusing features from two distinct modalities. We design an innovative multi-modal interaction and pooling attention network, which uses a small and flexible PAM module in the shallow layer of the network to enhance the feature extraction capability of the network and uses a MIM module in the last layer of the network to integrate RGB features and depth features effectively and then we design a RM during the upsampling stage for feature refinement. The network increases its focus on areas with more potential adjacent objects and overlaps, leading to improvement in the accuracy of RGB-D semantic segmentation. However, due to the attention mechanism adopted by our proposed network, the computational complexity of the network is relatively high. In future research, we will further optimize the network structure to reduce its computational complexity. In addition, we expect to further improve the accuracy of RGB-D segmentation by integrating multiple tasks such as depth estimation and semantic segmentation into a unified framework.

## Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html.

## Author contributions

SZ: Conceptualization, Formal Analysis, Methodology, Resources, Software, Validation, Visualization, Writing–original draft. MX: Data curation, Investigation, Supervision, Writing–review and editing.

## Funding

The authors declare that no financial support was received for the research, authorship, and/or publication of this article.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note



## References

1. Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Machine Intelligence* (2017) 39:640–51. doi:10.1109/tpami.2016.2572683

2. Li M, Wei M, He X, Shen F. Enhancing part features via contrastive attention module for vehicle re-identification. In: 2022 IEEE International Conference on Image Processing (ICIP); October 16-19, 2022; Bordeaux, France (2022). p. 1816–20.

3. Zhang Z. Microsoft kinect sensor and its effect. *IEEE MultiMedia* (2012) 19:4–10. doi:10.1109/mmul.2012.24

4. He Y, Chiu WC, Keuper M, Fritz M. Std2p: rgbd semantic segmentation using spatio-temporal data-driven pooling. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); July 21 2017 to July 26 2017; Honolulu, HI, USA (2017). p. 7158–67.

5. Couprie C, Farabet C, Najman L, LeCun Y *Indoor semantic segmentation using depth information* (2013). *arXiv preprint arXiv:1301.3572*.

6. Gupta S, Girshick R, Arbeláez P, Malik J. Learning rich features from rgb-d images for object detection and segmentation. *Computer Vision–ECCV 2014: 13th Eur Conf Zurich, Switzerland, September 6-12, 2014, Proc Part VII* (2014) 13:345–60. doi:10.1007/978-3-319-10584-0_23

7. Park SJ, Hong KS, Lee S. Rdfnet: rgb-d multi-level residual feature fusion for indoor semantic segmentation. In: Proceedings of the IEEE international conference on computer vision; 22-29 October 2017; Venice, Italy (2017). p. 4990–9.

8. Lee S, Park SJ, Hong KS. Rdfnet: rgb-d multi-level residual feature fusion for indoor semantic segmentation. In: 2017 IEEE International Conference on Computer Vision (ICCV); 22-29 October 2017; Venice, Italy (2017). p. 4990–9.

9. Eigen D, Fergus R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: 2015 IEEE International Conference on Computer Vision (ICCV); 7-13 December 2015; Santiago, Chile (2015). p. 2650–8.

10. Wang A, Lu J, Wang G, Cai J, Cham TJ. Multi-modal unsupervised feature learning for rgb-d scene labeling. In: Computer Vision–ECCV 2014: 13th European Conference; September 6-12, 2014; Zurich, Switzerland (2014). p. 453–67.

11. Shu Z, Li L, Yu J, Zhang D, Yu Z, Wu XJ. Online supervised collective matrix factorization hashing for cross-modal retrieval. *Appl intelligence* (2023) 53:14201–18. doi:10.1007/s10489-022-04189-6

12. Bai Y, Shu Z, Yu J, Yu Z, Wu XJ. Proxy-based graph convolutional hashing for cross-modal retrieval. *IEEE Trans Big Data* (2023) 1–15. doi:10.1109/tbdata.2023.3338951

13. Shu Z, Li B, Mao C, Gao S, Yu Z. Structure-guided feature and cluster contrastive learning for multi-view clustering. *Neurocomputing* (2024) 582:127555. doi:10.1016/j.neucom.2024.127555

14. Li L, Shu Z, Yu Z, Wu XJ. Robust online hashing with label semantic enhancement for cross-modal retrieval. *Pattern Recognition* (2024) 145:109972. doi:10.1016/j.patcog.2023.109972

15. Shu Z, Yong K, Yu J, Gao S, Mao C, Yu Z. Discrete asymmetric zero-shot hashing with application to cross-modal retrieval. *Neurocomputing* (2022) 511:366–79. doi:10.1016/j.neucom.2022.09.037

16. Yang J, Bai L, Sun Y, Tian C, Mao M, Wang G. Pixel difference convolutional network for rgb-d semantic segmentation. *IEEE Trans Circuits Syst Video Tech* (2024) 34:1481–92. doi:10.1109/tcsvt.2023.3296162

17. Zhao Q, Wan Y, Xu J, Fang L. Cross-modal attention fusion network for rgb-d semantic segmentation. *Neurocomputing* (2023) 548:126389. doi:10.1016/j.neucom.2023.126389

18. Yang E, Zhou W, Qian X, Lei J, Yu L. Drnet: dual-stage refinement network with boundary inference for rgb-d semantic segmentation of indoor scenes. *Eng Appl Artif Intelligence* (2023) 125:106729. doi:10.1016/j.engappai.2023.106729

19. Liu F, Shen C, Lin G. Deep convolutional neural fields for depth estimation from a single image. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 7 2015 to June 12 2015; Boston, MA, USA (2015). p. 5162–70.

20. Hu J, Huang Z, Shen F, He D, Xian Q. A bag of tricks for fine-grained roof extraction. *IGARSS 2023 - 2023 IEEE Int Geosci Remote Sensing Symp* (2023) 678–80. doi:10.1109/igarss52108.2023.10283210

21. Hu J, Huang Z, Shen F, He D, Xian Q. A rubust method for roof extraction and height estimation. In: IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium; 16 - 21 July, 2023; Pasadena, California, USA (2023). p. 770–1.

22. Hazirbas C, Ma L, Domokos C, Cremers D. Fusenet: incorporating depth into semantic segmentation via fusion-based cnn architecture. *Computer Vis – ACCV* (2017) 2016:213–28. doi:10.1007/978-3-319-54181-5_14

23. Hu X, Yang K, Fei L, Wang K. Acnet: attention based network to exploit complementary features for rgbd semantic segmentation. In: 2019 IEEE International Conference on Image Processing (ICIP); 22-25 September 2019; Taipei, Taiwan (2019). p. 1440–4.

24. Seichter D, Köhler M, Lewandowski B, Wengefeld T, Gross HM. Efficient rgb-d semantic segmentation for indoor scene analysis. In: 2021 IEEE International Conference on Robotics and Automation (ICRA); 30 May - 5 June 2021; Xian, China (2021). p. 13525–31.

25. Fu K, Fan DP, Ji GP, Zhao Q, Shen J, Zhu C. Siamese network for rgb-d salient object detection and beyond. *IEEE Trans Pattern Anal Machine Intelligence* (2022) 44: 5541–59. doi:10.1109/tpami.2021.3073689

26. Zhang X, Zhang S, Cui Z, Li Z, Xie J, Yang J. Tube-embedded transformer for pixel prediction. *IEEE Trans Multimedia* (2023) 25:2503–14. doi:10.1109/tmm.2022.3147664

27. Chen LZ, Lin Z, Wang Z, Yang YL, Cheng MM. Spatial information guided convolution for real-time rgbd semantic segmentation. *IEEE Trans Image Process* (2021) 30:2313–24. doi:10.1109/tip.2021.3049332

28. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Adv Neural Inf Process Syst* (2017) 30. doi:10.48550/ARXIV.1706.03762

29. Shen F, Wei M, Ren J *Hsgnet: object re-identification with hierarchical similarity graph network* (2022). arXiv preprint arXiv:2211.05486.

30. Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z, et al. Dual attention network for scene segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); June 18 2022 to June 24 2022; New Orleans, LA, USA (2019). p. 3141–9.

31. Shen F, Zhu J, Zhu X, Huang J, Zeng H, Lei Z, et al. An efficient multiresolution network for vehicle reidentification. *IEEE Internet Things J* (2022) 9:9049–59. doi:10.1109/jiot.2021.3119525

32. Shen F, Peng X, Wang L, Hao X, Shu M, Wang Y. Hsgm: a hierarchical similarity graph module for object re-identification. In: 2022 IEEE International Conference on Multimedia and Expo (ICME); July 18 2022 to July 22 2022; Taipei, Taiwan (2022). p. 1–6.

33. Woo S, Park J, Lee JY, Kweon IS. Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV); September 8-14, 2018; Munich, Germany (2018). p. 3–19.

34. Zhang Y, Wang Y, Li H, Li S. Cross-compatible embedding and semantic consistent feature construction for sketch re-identification. In: Proceedings of the 30th ACM International Conference on Multimedia (MM'22); October 10-14, 2022; Lisboa, Portugal (2022). p. 3347–55.

35. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, et al. Residual attention network for image classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); July 21 2017 to July 26 2017; Honolulu, HI, USA (2017). p. 6450–8.

36. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; June 18 2018 to June 23 2018; Salt Lake City, UT, USA (2018). p. 7132–41.

37. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q. Eca-net: efficient channel attention for deep convolutional neural networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); June 13 2020 to June 19 2020; Seattle, WA, USA (2020). p. 11531–9.

38. Qiao C, Shen F, Wang X, Wang R, Cao F, Zhao S, et al. A novel multi-frequency coordinated module for sar ship detection. In: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI); Oct. 31 2022 to Nov. 2 2022; Macao, China (2022). p. 804–11.

39. Ding M, Wang Z, Sun J, Shi J, Luo P. Camnet: coarse-to-fine retrieval for camera re-localization. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); Oct. 27 2019 to Nov. 2 2019; Seoul, Korea (2019). p. 2871–80.

40. Huang Z, Wang X, Wei Y, Huang L, Shi H, Liu W, et al. Ccnet: criss-cross attention for semantic segmentation. *IEEE Trans Pattern Anal Machine Intelligence* (2023) 45:6896–908. doi:10.1109/tpami.2020.3007032

41. Li H, Cen Y, Liu Y, Chen X, Yu Z. Different input resolutions and arbitrary output resolution: a meta learning-based deep framework for infrared and visible image fusion. *IEEE Trans Image Process* (2021) 30:4070–83. doi:10.1109/tip.2021.3069339

42. Li H, Liu J, Zhang Y, Liu Y. A deep learning framework for infrared and visible image fusion without strict registration. *Int J Comp Vis* (2023) 132:1625–44. doi:10.1007/s11263-023-01948-x

43. Li H, Zhao J, Li J, Yu Z, Lu G. Feature dynamic alignment and refinement for infrared–visible image fusion:translation robust fusion. *Inf Fusion* (2023) 95:26–41. doi:10.1016/j.inffus.2023.02.011

44. Xiao W, Zhang Y, Wang H, Li F, Jin H. Heterogeneous knowledge distillation for simultaneous infrared-visible image fusion and super-resolution. *IEEE Trans Instrumentation Meas* (2022) 71:1–15. doi:10.1109/tim.2022.3149101

45. Xiang K, Yang K, Wang K. Polarization-driven semantic segmentation via efficient attention-bridged fusion. *Opt Express* (2021) 29:4802–20. doi:10.1364/oe.416130

46. Shen F, Zhu J, Zhu X, Xie Y, Huang J. Exploring spatial significance via hybrid pyramidal graph network for vehicle re-identification. *IEEE Trans Intell Transportation Syst* (2022) 23:8793–804. doi:10.1109/tits.2021.3086142

47. Shen F, Xie Y, Zhu J, Zhu X, Zeng H. Git: graph interactive transformer for vehicle re-identification. *IEEE Trans Image Process* (2023) 32:1039–51. doi:10.1109/tip.2023.3238642

48. Zhuang Z, Li R, Jia K, Wang Q, Li Y, Tan M. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); Oct. 11 2021 to Oct. 17 2021; Montreal, BC, Canada (2021). p. 16260–70.

49. Zhu Z, He X, Qi G, Li Y, Cong B, Liu Y. Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal mri. *Inf Fusion* (2023) 91: 376–87. doi:10.1016/j.inffus.2022.10.022

50. Zhu Z, Sun M, Qi G, Li Y, Gao X, Liu Y. Sparse dynamic volume transunet with multi-level edge fusion for brain tumor segmentation. *Comput Biol Med* (2024) 172: 108284. doi:10.1016/j.compbiomed.2024.108284

51. Liu Y, Shi Y, Mu F, Cheng J, Chen X. Glioma segmentation-oriented multi-modal mr image fusion with adversarial learning. *IEEE/CAA J Automatica Sinica* (2022) 9: 1528–31. doi:10.1109/jas.2022.105770

52. Liu Y, Mu F, Shi Y, Chen X. Sf-net: a multi-task model for brain tumor segmentation in multimodal mri via image fusion. *IEEE Signal Process. Lett* (2022) 29:1799–803. doi:10.1109/lsp.2022.3198594

53. Silberman N, Hoiem D, Kohli P, Fergus R. Indoor segmentation and support inference from rgbd images. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision; October 7-13, 2012; Florence, Italy (2012). p. 746–60.

54. Song S, Lichtenberg SP, Xiao J. Sun rgb-d: a rgb-d scene understanding benchmark suite. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 7 2015 to June 12 2015; Boston, MA, USA (2015). p. 567–76.

55. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 27 2016 to June 30 2016; Las Vegas, NV, USA (2016). p. 770–8.

56. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis* (2015) 115:211–52. doi:10.1007/s11263-015-0816-y

57. Fu X, Shen F, Du X, Li Z. Bag of tricks for "vision meet alage" object detection challenge. In: 2022 6th International Conference on Universal Village (UV); October 19-22, 2024; Boston, USA (2022). p. 1–4.

58. Shen F, He X, Wei M, Xie Y *A competitive method to vipriors object detection challenge* (2021). arXiv preprint arXiv:2104.09059.

59. Shen F, Wang Z, Wang Z, Fu X, Chen J, Du X, et al. *A competitive method for dog nose-print re-identification* (2022). arXiv preprint arXiv:2205.15934.

60. Xu X, Liu J, Liu H. Interactive efficient multi-task network for rgb-d semantic segmentation. *Electronics* (2023) 12:3943. doi:10.3390/electronics12183943

61. Zhang Y, Xiong C, Liu J, Ye X, Sun G. Spatial information-guided adaptive context-aware network for efficient rgb-d semantic segmentation. *IEEE Sensors J* (2023) 23:23512–21. doi:10.1109/jsen.2023.3304637

62. Wu Z, Allibert G, Stolz C, Ma C, Demonceaux C *Depth-adapted cnns for rgb-d semantic segmentation* (2022). arXiv preprint arXiv:2206.03939.

63. Xue Z, Marculescu R. Dynamic multimodal fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; June 18 2022 to June 24 2022; New Orleans, LA, USA (2023). p. 2575–84.

64. Yan X, Hou S, Karim A, Jia W. Rafnet: rgb-d attention feature fusion network for indoor semantic segmentation. *Displays* (2021) 70:102082. doi:10.1016/j.displa.2021.102082

65. Chen X, Lin KY, Wang J, Wu W, Qian C, Li H, et al. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In: European Conference on Computer Vision; 23-28 August; Glasgow, United Kingdom (2020). p. 561–77.

66. Cao J, Leng H, Lischinski D, Cohen-Or D, Tu C, Li Y. Shapeconv: shape-aware convolutional layer for indoor rgb-d semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision; Oct. 11 2021 to Oct. 17 2021; Montreal, BC, Canada (2021). p. 7068–77.

67. Seichter D, Fischedick SB, Köhler M, Groß HM. Efficient multi-task rgb-d scene analysis for indoor environments. In: 2022 International Joint Conference on Neural Networks (IJCNN); 18-23 July 2022; Padua, Italy (2022). p. 1–10.

68. Tang Q, Liu F, Zhang T, Jiang J, Zhang Y. Attention-guided chained context aggregation for semantic segmentation. *Image Vis Comput* (2021) 115:104309. doi:10.1016/j.imavis.2021.104309

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 11 June 2024
DOI 10.3389/fphy.2024.1417536



OPEN ACCESS

EDITED BY
Zhenqiu Shu,
Kunming University of Science and Technology, China

REVIEWED BY
Huajin Li,
Chengdu University, China
Kang Liao,
Southwest Jiaotong University, China

*CORRESPONDENCE
Rubin Wang,
rbwang_hhu@foxmail.com
Yipeng Lei,
yipenglei@163.com
Yue Yang,
youngy0528@163.com

RECEIVED 15 April 2024
ACCEPTED 14 May 2024
PUBLISHED 11 June 2024

CITATION
Wang R, Lei Y, Yang Y, Xu W and Wang Y (2024), Dynamic prediction model of landslide displacement based on (SSA-VMD)-(CNN-BiLSTM-attention): a case study.
*Front. Phys.* 12:1417536.
doi: 10.3389/fphy.2024.1417536



# Dynamic prediction model of landslide displacement based on (SSA-VMD)-(CNN-BiLSTM-attention): a case study

Rubin Wang[1,2]*, Yipeng Lei[2]*, Yue Yang[2]*, Weiya Xu[1,2] and Yunzi Wang[2]

[1]Key Laboratory of Ministry of Education for Geomechanics and Embankment Engineering, Hohai University, Hohai, China, [2]Research Institute of Geotechnical Engineering University, Nanjing, China

Accurately predicting landslide displacement is essential for reducing and managing associated risks. To address the challenges of both under-decomposition and over-decomposition in landslide displacement analysis, as well as the low predictive accuracy of individual models, this paper proposes a novel prediction model based on time series theory. This model integrates a Convolutional Neural Network (CNN) with a Bidirectional Long-Short Term Memory network (BiLSTM) and an attention mechanism to form a comprehensive CNN-BiLSTM-Attention model. It harnesses the feature extraction capabilities of CNN, the bidirectional data mining strength of BiLSTM, and the focus-enhancing properties of the attention mechanism to enhance landslide displacement predictions. Furthermore, this paper proposes utilizing the Variational Mode Decomposition (VMD) method to decompose both landslide displacement and its influencing factors. The VMD algorithm's parameters are optimized through the Sparrow Search Algorithm (SSA), which effectively minimizes the influence of subjective bias while maintaining the integrity of the decomposition. A fusion of the Maximal Information Coefficient (MIC) and Grey Relational Analysis (GRA) is then employed to identify the critical influencing factors. The selected sequence of factors that conforms to the criteria is used as the input variable for displacement prediction via the CNN-BiLSTM-Attention model. The cumulative displacement prediction is derived by aggregating the results from each sequence. The study reveals that the SSA-VMD-CNN-BiLSTM-Attention model introduced herein achieves superior predictive accuracy for both periodic and random term displacements than individual models. This advancement provides a dependable benchmark for forecasting displacement in similar landslide scenarios.

KEYWORDS

**landslide displacement prediction model, variational mode decomposition, maximal information coefficient, bidirectional long short term memory network, attention mechanism**

## 1 Introduction

Landslides are frequent and destructive geological disasters in China, posing constant threats to the safety of nearby villagers. The deformation evolution of landslides is a complex nonlinear system influenced by both intrinsic geological conditions and external environmental factors. [1]. Displacement serves as a direct indicator of the progression

trends and kinematic patterns of landslides. A thorough analysis of landslide displacement is crucial for accurately identifying the evolutionary stages of landslides, effectively mitigating disaster risks, and minimizing losses. [2, 3].

Currently, scholars typically decompose landslide displacement sequences using time series theory and construct prediction models to forecast displacement component [4]. Commonly used displacement decomposition methods include the moving average method [5, 6], empirical mode decomposition (EMD) [7–9], ensemble empirical mode decomposition (EEMD) [10–12], wavelet transform (WT) [13–15] and Variational Mode Decomposition (VMD) [16–20]. Although the methods mentioned above have yielded positive outcomes in decomposing displacement sequences, they also have their limitations. For instance, while the moving average method is clear physical interpretation, it cannot decompose the random term displacement. Although EMD, EEMD, and WT have addressed the limitations of the moving average method, the number of decomposed sequences is uncontrollable, and the physical meaning of each component is unclear. Furthermore, it should be noted that WT and Discrete Wavelet Transform (DWT) differ in their approach to determining basis functions and wavelet orders. VMD, on the other hand, addresses the issue of modal aliasing and allows for the specification of the number of components after decomposition, with each component having a clear physical interpretation. However, the effectiveness of the decomposition and the fidelity of the results depend heavily on the selection of parameters. To fully utilize the benefits of the VMD algorithm, which has high adaptability and clear physical meaning for each component, this paper optimizes the penalty factor $\alpha$ and the rise time step $\tau$ in the VMD model using the Sparrow Search Algorithm (SSA). The VMD decomposition effect and fidelity are measured using the sample entropy of the periodic term displacement or the low frequency of the influencing factor as the root mean square error of the original displacement and the reconstructed displacement.

The construction of a prediction model plays a pivotal role in determining the precision of landslide displacement forecasts. Models for predicting landslide displacement can be categorized into three types: historical experience models, statistical models, and machine learning models. The empirical model based on historical experience requires a significant amount of data and experimentation to verify its accuracy and has strict application conditions. Although the statistical model is effective in monitoring landslides influenced by a single factor, its ability to consider and predict the impact of multiple factors is often limited. As computer technology advances rapidly, machine learning models have become increasingly prevalent for predicting landslide displacement. These models, with their straightforward calculation procedures, accurate prediction outcomes, and low computational requirements, are adept at managing nonlinear relationships. Machine learning models are increasingly popular for predicting landslide displacement. Due to their simple calculation processes, accurate prediction results, low computational costs, and ability to handle nonlinear relationships [21], machine learning models are widely employed for landslide displacement prediction. For instance, Yang et al. [22] employed support vector machines (SVM) to predict landslide displacement. However, the prediction error for individual points was significant. Du et al. [23] established a neural network model for predicting landslide displacement based on the analysis of inducing factors. Wang et al. [24] developed a prediction model for landslide displacement by combining the Extreme Learning Machine (ELM) with Random Search Support Vector Regression (RS-SVR) sub-models. Li et al. [25] proposed an ensemble-based extreme learning approach to study landslide displacement prediction. The results demonstrated that the integrated model achieved higher prediction accuracy compared to a single model. Wang et al. [26] studied and compared the predictive capabilities of reservoir landslide displacement using various machine learning approaches. Relying solely on individual prediction accuracy to assess the superiority of machine learning methods may not be reliable, whereas the combined model offers improved average prediction accuracy and predictive stability. However, the model does not fully consider the dynamic characteristics of landslide evolution. This is because the evolution process of a landslide is inherently a dynamic system, and treating it as a static regression problem reduces the accuracy of displacement predictions [27, 28]. Accurate prediction of landslide displacement necessitates a dynamic prediction model capable of simulating the changes in landslide displacement. Li et al. [28] proposed a modeling and prediction framework for landslide displacement based on a deep belief network and the exponentially weighted moving average (EWMA) control chart, obtaining excellent prediction results. The Long Short-Term Memory (LSTM) model is a type of dynamic neural network that integrates delay units and feedback into the static network, enhancing its sensitivity to historical factors and output. This trait renders it more suitable for predicting landslide displacement influenced by multiple factors.

The LSTM model is a dynamic modelling method commonly used to predict landslide displacement [29, 30]. Previous studies have demonstrated that the prediction accuracy of LSTM is superior to that of backpropagation neural network, ELM or SVM [31]. However, the LSTM model relies exclusively on past state information, which qualifies it as a unidirectional network. The bidirectional LSTM (BiLSTM) network is an enhancement and expansion of the traditional LSTM. It can increase its predictive accuracy by learning input time series data from both forward and backward directions, as noted in references [32–34]. More recently, the progression of deep neural networks has given rise to stable and highly accurate models for data processing and industrial predictions, such as the convolutional neural network (CNN) and BiLSTM. The combined CNN-BiLSTM model merges CNN's feature learning capabilities with BiLSTM's time series memory function, resulting in further improvements in prediction accuracy and operational efficiency [35,36]. Nava et al. [37] used seven different machine learning models to predict four types of landslide displacement, taking into account various geographic locations, geological settings, time intervals, and measurement instruments. The results indicated that deep learning ensemble models surpassed others in performance, especially for the seasonal Baishuihe landslide. Lin et al. [38] proposed a combined model based on the CNN-BiLSTM framework. This model demonstrated higher prediction accuracy when compared to both the traditional LSTM model and the CNN-LSTM combined model. Wang et al. [39] applied the CNN-LSTM model to dynamically predict landslide displacement, finding that the CNN-BiLSTM model's prediction accuracy exceeded that of BP, LSTM, and

GRU models. However, the deep learning methods mentioned previously fall short when handling multi-dimensional feature data, such as in predicting landslide displacement, due to the absence of an effective weighted input feature mechanism. Not all input features equally influence landslide deformation; certain factors may contribute minimally to the prediction of landslide displacement. An excessively large proportion of feature weights could compromise the prediction model's accuracy.

In recent years, attention mechanisms have become increasingly prominent in image recognition and machine translation. These mechanisms function as an effective resource allocation system by assigning differential weights to input features in order to emphasize the most significant information [40]. Tang et al. [41] applied a BiLSTM model with an attention mechanism to predict landslide displacement, and it was found that this combination yielded better results than using the traditional LSTM model alone. Furthermore, it's common for researchers to rely on correlation evaluation methods to select input variables for prediction models. However, this method may result in one-sided evaluations and the inclusion of irrelevant data, which can increase computational complexity and reduce prediction accuracy.

To summarize, this article uses the BaiShuihe landslide in the Three Gorges Reservoir area as an example. It first applies the SSA-VMD model to deconstruct the landslide displacement sequence into trend term displacement, periodic term displacement, and random term displacement, while simultaneously decomposing the triggering factors into high-frequency and low-frequency parts. Next, it employs a fusion technique that combines the Maximal Information Coefficient and Grey Relation Analysis (MIC-GRA) to filter the influencing factors of landslide displacement from different angles. Finally, the CNN-BiLSTM-Attention composite model is utilized to predict the various displacement components. The predicted trend, periodic term, and random term displacements are then aggregated and reconstructed, with an evaluation and analysis of the results following. The forecasting performance has been confirmed, and the insights from this study establish a robust foundation for the future development of landslide displacement prediction and early warning systems.

# 2 Methodology

## 2.1 Displacement time series additive model

Predicting time series data presents a significant challenge in the field of statistical analysis, especially when employing time series analysis methods. Previous studies [5, 10, 17, 19, 30] have documented that the cumulative displacement of landslides is a complex, nonlinear sequence. Time series analysis facilitates the decomposition of cumulative displacement into three distinct segments. Predominantly, landslide deformation is influenced by trend term displacement, which arises from internal geological conditions such as the topography, geological structure, and strata lithology. The trend term displacement, which is influenced by internal factors, can be represented as an approximately monotonic increasing function over time [5, 11, 16]. This paper explores the impact of time on the trend term, along with the periodic and random term displacements. The periodic term displacement arises through the collective effects of external factors such as rainfall and reservoir water levels, resulting in displacement that typically exhibits an approximate periodic pattern, as identified in earlier studies [17, 19, 21]. Meanwhile, the random term displacement is attributed to stochastic factors including wind load, vehicular load, and seismic activity, as documented in the literature [22, 29]. The cumulative displacement of landslides, according to the findings from time series additive model research, can be expressed as Eq. 1:

$$X(t) = T(t) + P(t) + R(t) \tag{1}$$

where $X(t)$ is the displacement value of the time series, $T(t)$ is a trend term function, $P(t)$ is a periodic term function, and $R(t)$ is an random term function with uncertainty.

## 2.2 Specific steps of variational mode decomposition

In 2014, K. Dragomiretskiy and D. Zosso proposed the variational mode decomposition (VMD) as an adaptive, non-recursive method for signal processing based on the EMD model [42]. The VMD decomposes a real input signal into multiple Intrinsic Mode Function (IMF) components with specific sparse characteristics. This approach determines the number of modal components in advance, overcoming the endpoint effects and modal component aliasing problems of EMD methods. Furthermore, it can decrease the non-stationary nature of time series data with high complexity and strong nonlinearity, leading to subsequences with distinct sparse features. The equation for the IMF in VMD is a form of amplitude modulation frequency modulation signal $u_k$(t), which is expressed as Eq. 2:

$$u_k(t) = A_k(t)\cos\left(\phi_k(t)\right) \tag{2}$$

where $\phi_k(t)$ is the phase, $A_k(t)$ is the instantaneous amplitude, $\phi_k(t)$ is the non decreasing function, and $A_k(t)$ is consistent with the mean positive number.

The sum of the input signal sequence and modes is used as the constrained variational expression. The constrained variational expression is written as Eqs 3 and 4:

$$\min_{\{u_k\},(w_k)} \left\{ \sum_k \left\| \partial_t \left[ \left(\delta(t) + j/\pi t\right) * u_k\left(e^{-j\omega_k t}\right) \right] \right\|_2^2 \right\} \tag{3}$$

$$\sum_k^K u_k = x_t \tag{4}$$

where $K$ is the required number of modal components, which is an integer between one and $K$. $\{u_k\} = \{u_1, \cdots, u_k\}$ is the modal component obtained from the final decomposition. $\{\omega_k\} = \{\omega_1, \cdots, \omega_k\}$ is the actual center frequency of each modal component. $\partial_t$ is a partial derivative symbol. $\delta(t)$ is the Dirac function. * is a convolution operator.

To solve the equation above, we introduce the Lagrange operator λ to transform the constrained variational problem into an unconstrained one. The extended Lagrange expression is obtained as Eq. 5:

$$L(\{u_k\},\{\omega_k\},\lambda) = \varepsilon \sum_k \left\| \partial_t \left[ (\delta(t) + j/\pi t) * u_k\left(e^{-j\omega_k t}\right) \right] \right\|_2^2 + \left\| x(t) - \sum_k u_k(t) \right\|_2^2 + \langle \lambda(t), x(t) - \sum_k u_k(t) \rangle \tag{5}$$

where $\varepsilon$ is used to decompose and reduce the interference of Gaussian noise. The optimal solution of the constrained model can be obtained by using the alternating direction multiplier iterative algorithm to optimize the modal components and center frequencies, and searching for the saddle points of the unconstrained model, thereby obtaining $K$ modal components. The aim of this study was to decompose landslide displacement and influencing factors using VMD. The time series additive model of landslide displacement was used to set the number of modal components $K = 3$. The influence factor time series $K = 2$ modal components. The low frequency component of the influence factor mainly affected the periodic displacement of the landslide, while the high-frequency component contributed to the random displacement [19]. Utilizing the VMD algorithm to dissect landslide displacement into three components, it is pivotal to recognize that the outcomes might not carry practical or tangible physical relevance. The parameters $\alpha$ and $K$ have been determined, and they will affect the decomposition effect and fidelity. Efficient and accurate selection of parameters in the VMD algorithm will be crucial for the decomposition of displacement time series. The SSA was chosen to optimize the penalty function $\alpha$ and rise time step $\tau$ in the VMD model. This approach effectively avoids the influence of subjective factors.

## 2.3 Variational modal decomposition for the sparrow optimization algorithm

### 2.3.1 Sample entropy

Sample entropy is a complexity metric for time series analysis, proposed by Richman [43] in response to the limitations encountered with approximate entropy. This measure effectively mitigates deviations arising from template matching issues, by considering the probability and complexity of emergent patterns within a time series. Contrary to approximate entropy, Sample entropy maintains independence from the length of the sequence, yielding higher consistency across analyses. This attribute renders it an essential tool for researchers and practitioners seeking to accurately gauge the intricacies of time series data.

For a given time series $\{x(t)\}, t = 1, 2, \cdots, N$ with length $N$, the sample entropy calculation steps of the time series are as follows:

(1) The $m$-dimensional vector $\{x^m(t)\}, t = 1, 2, \cdots, N - m + 1$ is constructed at time $t$, where $m$ is the embedding dimension of the vector. The distance between the time series is defined as the absolute value of the maximum difference between the elements of the two sub-sequences is $d_{ij}^m$, and the calculation formula is as Eq. 6:

$$\begin{aligned} d_{ij}^m &= d\left[x_i^m, x_j^m\right] \\ &= \max_{k=0,1\cdots m-1} |x(i+k) - x(i-k)|, (i, j, \cdots, N-m+1, \text{且} i \neq j) \end{aligned} \tag{6}$$

(2) Setting the similarity tolerance $r\,(r > 0)$, and calculating the number ratio of the distance between $x_i^m$ and $x_j^m$ less than $r$, denoted as $B_i^m(r)$, and the calculation formula is as Eq. 7:

$$B_i^m(r) = \frac{num\left\{d_{ij}^m < r\right\}}{N - m + 1} \tag{7}$$

where $num\{\cdot\}$ is the counting function. By calculating the number of vectors whose distance between $x_i^m$ and $x_j^m$ is less than $r$, the formula for calculating the average template match probability $B^r(r)$ is as Eq. 8:

$$B^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} B_i^m(r) \tag{8}$$

(3) The $m + 1$ dimensional sequence is constructed, and the average template match probability $B^{m+1}(r)$ with a distance less than $r$ between $x_i^m$ and $x_j^m$ is calculated by repeating the Eqs 7 and 8, where $B^m(r)$ and $B^{m+1}(r)$ are the probabilities of $m$ and $m + 1$ points respectively under the condition of similar tolerance $r$, respectively. The sampling entropy of $\{x(i)\}$ is defined as Eq. 9:

$$SampEn(m, r) = \lim_{N \longrightarrow \infty} \left\{ -\ln\left[\frac{B^{m+1}(r)}{B^m(r)}\right] \right\} \tag{9}$$

When the length of the time series is finite, the sample entropy can be calculated as Eq. 10:

$$SampEn(m, r, N) = -\ln\left[\frac{B^{m+1}(r)}{B^m(r)}\right] \tag{10}$$

where $m$ is the embedding dimension, generally taken as one or 2. $r$ is the similarity tolerance, generally taken as $0.1 \sim 0.25\sigma_x$, and $\sigma_x$ is the standard deviation of the sequence. The sample entropy value increases with the complexity of the time series and decreases with its simplicity. This paper uses the sample entropy of the decomposed periodic term displacement sequence as an indicator to evaluate the decomposition effect of the VMD algorithm. A smaller entropy value of periodic term displacement indicates a better decomposition effect.

### 2.3.2 Basic principles of sparrow optimization algorithm

The Sparrow Search Algorithm (SSA) is a population-based intelligent optimization algorithm introduced by Xue et al. [44]. The algorithm derives its optimization strategy from the foraging and anti-predation behavior observed in sparrows. As a swarm intelligence algorithm, it outperforms many others in terms of search precision, convergence speed, stability, and resilience. Its successful applications span a range of problems in diverse domains, including workshop scheduling, parameter optimization, image classification, and graphical optimization tasks. Building on this success, the present article employs SSA to autonomously determine the optimal parameters for the penalty factor and the rise time step within the VMD algorithm.

The SSA categorizes sparrows into three roles during the search process: the discoverer, the follower, and the sentinel. Their positional updates are as Eqs 11-13:

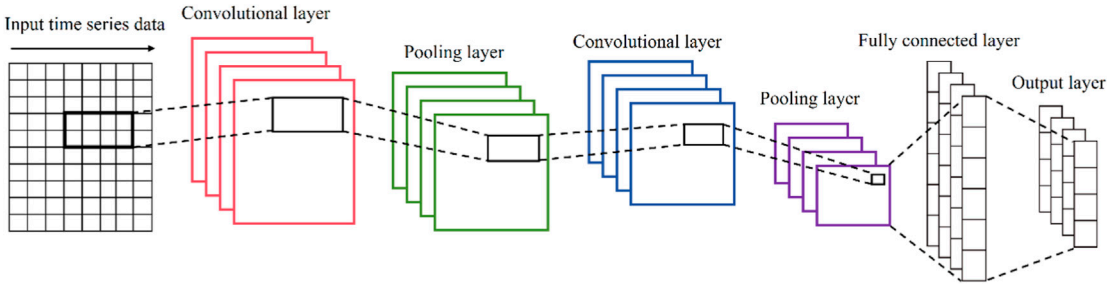



FIGURE 1
Two-layer one-dimensional CNN convolution structure.

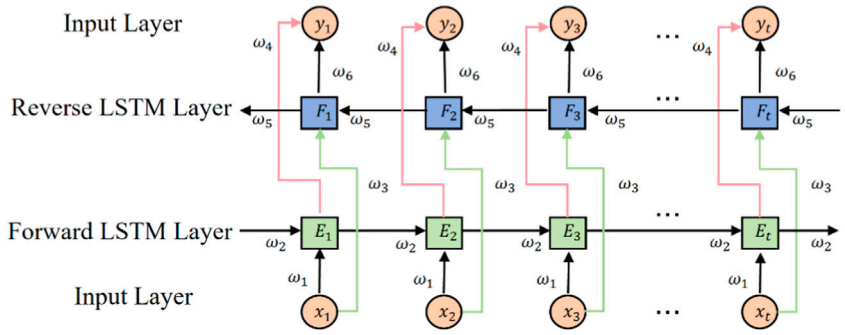



FIGURE 2
BiLSTM network structure diagram [53].

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \exp\left[-i/\alpha i_{ter\,max}\right], R_2 < S_T \\ X_{i,j}^{t} + QL, R_2 \geq S_T \end{cases} \tag{11}$$

$$X_{i,j}^{t+1} = \begin{cases} Q \exp\left[\left(X_{worst} - X_{i,j}^{t}\right)/i^2\right], i > n/2 \\ X_{p}^{t+1} + \left|X_{i,j}^{t} - X_{p}^{t+1}\right| A^{+} L, i \leq n/2 \end{cases} \tag{12}$$

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta\left|X_{i,j}^{t} - X_{best}^{t}\right|, f_i > f_g \\ X_{i,j}^{t} + K\left(\dfrac{\left|X_{i,j}^{t} - X_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right), f_i = f_g \end{cases} \tag{13}$$

where $t$ is the current number of iterations, $i_{ter\,max}$ is the maximum number of iterations, α is a uniform random number of $(0, 1]$, $Q$ is a standard normal distribution random number, $X_{i,j}$ is the position information of $i$ sparrow in $j$ dimension, $L$ is a matrix with all elements one, $R_2 \in [0, 1]$ is the warning value, $S_T \in [0.5, 1]$ is the warning threshold. $X_{worst}$ is the worst position in the global, $X_p^t$ is the optimal position occupied by the discoverer, $A$ is a multidimensional matrix of one or -1, $n$ is the number of sparrows. $X_{best}$ is the current global best position, $\beta$ is the control parameter for the step size, $K$ is a uniform random number between $[-1, 1]$, $K$ represents the direction of movement of the sparrow, $f_i$ is the fitness of the current sparrow, $f_g$ is the best fitness value of the global, $f_w$ is the worst fitness value, $\varepsilon$ is a small constant.

To optimize SSA, determining the fitness function is a crucial step. The fidelity of the decomposed VMD algorithm is evaluated by accumulating and reconstructing the decomposed subsequences into $m$. To measure the integrity of the decomposed sequence, the root mean square error (RMSE) between the reconstructed sequence and the original sequence M is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_t - \hat{x}_t)} \tag{14}$$

where $x_t$ is the value of the original sequence at time $t$, $\hat{x}_t$ is the value of the reconstructed sequence at time $t$, and $n$ is the length of the sequence.

Eq 14 demonstrates that a smaller RMSE value implies a smaller error between the reconstructed sequence $m$ and the original sequence $M$, indicating a reduced loss of the original sequence. This paper combines sample entropy and root mean square error to effectively reflect the completeness of the decomposed sequence and the success of the decomposition. The function expression is as Eq. 15:

$$fitness = RMSE(m, M) \cdot SampEn(IMF_2) \tag{15}$$

where $RMSE(m, M)$ is the root mean square error between the reconstructed sequence and the original sequence. $SampEn(IMF_2)$ is the sample entropy value of the low-frequency part of the periodic term displacement sequence or influencing factor after decomposition. The fitness value of the SSA algorithm is determined using the calculated value of Equation 15. To find

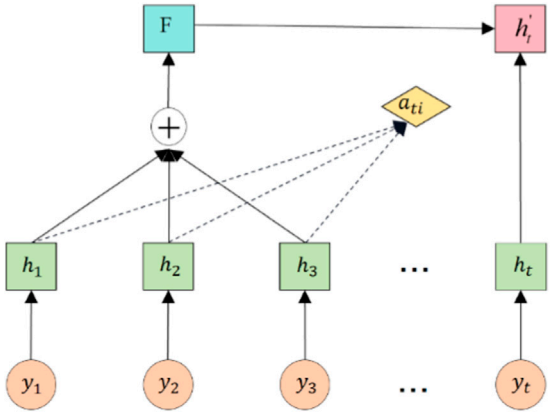



FIGURE 3
Attention mechanism structure diagram [53].

the optimal fitness, the penalty factor α and rise time step τ are optimized. The process is outlined in the following steps:

(1) Input displacement time series signal.
(2) Initialize the parameter input of the sparrow optimization algorithm, and randomly generate a series of α and τ as the initial position of the sparrow population.
(3) Perform VMD decomposition on the displacement sequence of the current sparrow position. Calculate the sample entropy of the decomposed periodic term sequence or low-frequency part of the influencing factors with confidence.
(4) The acclimatization value for each sparrow was calculated according to Eq 10, identify the optimal and worst fitness individuals, and update the positions of discoverers, followers, and early warning individuals according to Eqs 11–13.
(5) Repeat (3) and (4) until the maximum number of iterations is reached, and output the sparrow position and fitness values at this time as the optimal solution.

## 2.4 Maximal information coefficient

Mutual information (MI) [45] developed from Shannon entropy theory, is a method for analyzing the statistical correlation between two random variables. It is adept at detecting both linear and non-linear relationships among variables. Despite its utility, mutual information is not a normalized metric, which limits its capacity to provide a quantitative assessment of correlation strength. To address this limitation, this article introduces the Maximal Information Coefficient (MIC). Proposed by Reshef et al. in 2011 in the journal Science [46], MIC builds upon MI to evaluate the degree of dependency between variables comprehensively. It is competent in quantifying not only linear but also non-linear and non-functional correlations among variables.

The principle of the MIC is for a given two random variables $X, Y$ and a finite ordered data set $D(X,Y) = \{(x_i, y_i), i = 1, 2, \cdots, n\}$, the $X$ and $Y$ regions in $D$ are divided respectively into grids $x \times y$ of $G$. Then, the probability distribution of the data set $D$ on the grid $G$ is $D|_G$, and the mutual information value $I(D|_G)$ under this segmentation mode is calculated. Finally, the maximum mutual information value under all possible grid segmentations G is obtained as Eq. 16:

$$I^*(D,x,y) = \max I(D|_G) \tag{16}$$

By normalizing $I^*(D,x,y)$ function, the characteristic matrix element $I^*(D,x,y)$ of the variable can be obtained by Eq. 17:

$$M(D)_{x,y} = \frac{I^*(D,x,y)}{\log(\min\{x,y\})} \tag{17}$$

Different $x \times y$ values divide the grid to get different $M(D)_{x,y}$ values, and the maximum $M(D)_{x,y}$ is called the MIC of variable $Y$, and the maximum $M(D)_{x,y}$ is expressed as Eq. 18:

$$MIC(D) = \max_{xy<B(n)} M(D)_{x,y} \tag{18}$$

where $B(n)$ is the maximum number of meshes, $n$ is the capacity of the data sample and usually set to $B = n^{0.6}$ [47, 48], This paper also adopts this value.

## 2.5 Construction of CNN-BiLSTM-attention combination model

### 2.5.1 CNN principle structure

A Convolutional Neural Network (CNN) is the neural network model most frequently employed in deep learning [49]. Its potent feature-learning capability substantially diminishes the model's parameter count, which has led to its extensive application in image recognition and computer vision domains. Over recent years, a growing number of researchers have effectively utilized CNN for time series analysis. The model's distinct features, such as weight-sharing and localized connections, can significantly diminish the parameter quantity needed for training. These attributes facilitate faster model training velocities and allow for the more proficient extraction of features from the input data [50].

The CNN consists of a convolution layer, pooling layer, fully connected layer, and output layer. The convolution layer applies the activation function to perform non-linear operations on the input time series data and extract local feature information. The pooling layer uses a pooling function to decrease the dimensionality of the convolution output, and improve the model's robustness and generalization ability. The fully connected layer then maps the data output from the pooling layer to a fixed-length column vector. This paper uses a two-layer one-dimensional CNN convolution structure to extract feature information, as shown in Figure 1.

### 2.5.2 BiLSTM model

In 1997, Sepp Hochreiter et al. proposed long short-term memory networks (LSTM). The gate structure and internal memory unit effectively solve the problem of gradient disappearance and explosion in long sequence training of the RNN model [51]. The model's control unit consists of a forget gate, an input gate, and an output gate. The respective calculation formulas for these components are as Eqs 19-24:

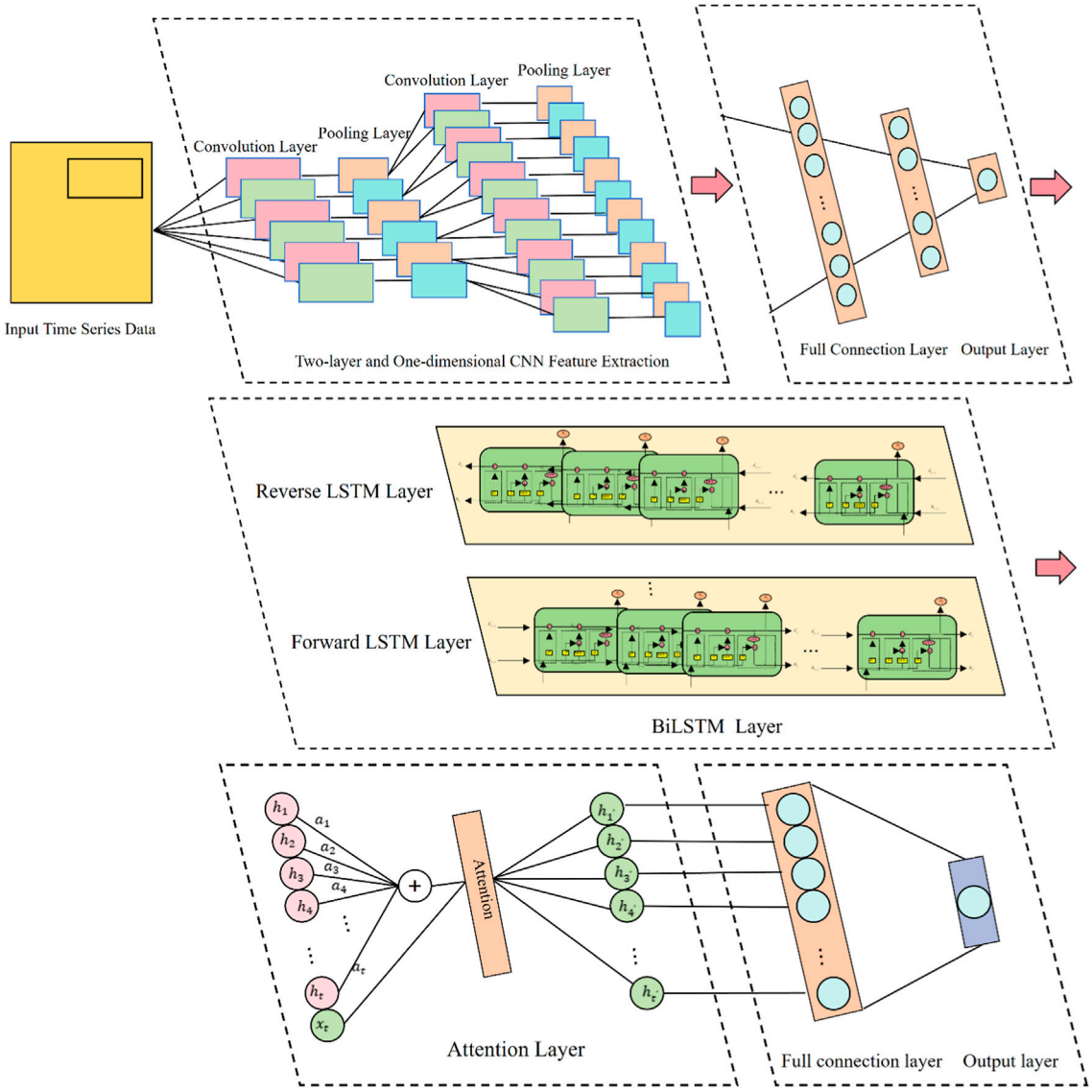



FIGURE 4
Diagram of CNN-BiLSTM-Attention network structure.

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{19}$$

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{20}$$

$$\tilde{C}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{21}$$

$$C_t = i_t \times \tilde{C}_t + f_t \times C_{t-1} \tag{22}$$

$$O_t = sigmoid\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{23}$$

$$h_t = O_t \times \tanh\left(C_t\right) \tag{24}$$

where $f_t$、$i_t$、$O_t$ denote the forgetting gate, the input gate and the output gate, respectively, $W_f$、$W_c$、$W_o$ denote the weights of the corresponding gates, $b_f$、$b_c$、$b_o$ denote the corresponding bias, $x_t$ denote the input time series data, $t$ denote the sigmoid activation function, and $\sigma$ is the hyperbolic tangent activation function, $C_t$ and $\tilde{C}_t$ denote the cell state and temporary state of the cell, respectively.

The Bidirectional Long Short-Term Memory (BiLSTM) model significantly enhances the traditional LSTM model. By leveraging both forward and reverse LSTM processes, it effectively integrates information from both past and future contexts, enabling it to make more accurate predictions. Consequently, it outperforms the LSTM model in prediction accuracy [33,34,52]. The structure of the BiLSTM model is depicted in Figure 2.

### 2.5.3 Attention mechanism

The Attention mechanism allocates weights to different features, assigning greater weights to key content and smaller weights to other content. This allocation improves the efficiency of information processing and the prediction accuracy of the model [54]. The Attention unit structure is displayed in the Figure 3. The formula of attention mechanism can be referred to [53].

### 2.5.4 Prediction process of CNN-BiLSTM-attention combined model

This paper presents a dynamic displacement prediction method based on the CNN-BiLSTM-Attention model. The model utilizes a

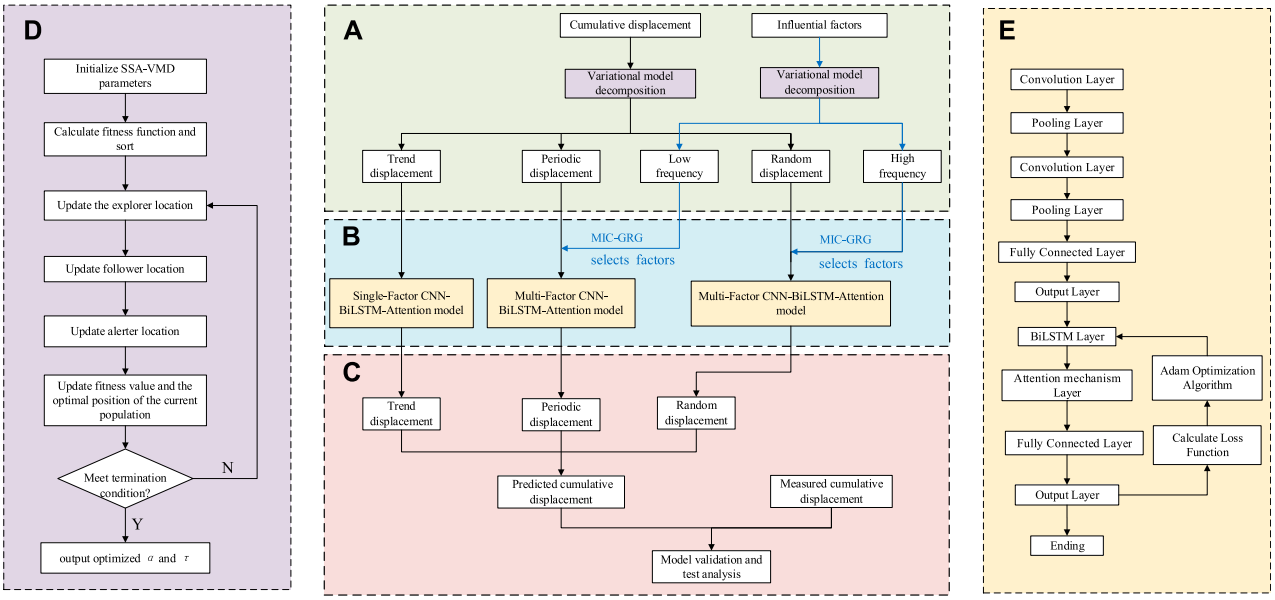



FIGURE 5
Flowchart of the displacement prediction. **(A)**. Landslide displacement decomposition and data set establishment; **(B)**. Training of CNN-BiLSTM-Attention model; **(C)**. Verification of CNN-BiLSTM-Attention model; **(D)**. SSA optimize the parameters of VMD; **(E)**. CNN-BiLSTM-Attention model establishment.

CNN framework comprising of a two-layer one-dimensional convolutional layer and a pooling layer to automatically extract the internal features of the displacement sequence. The convolutional layer efficiently performs nonlinear local feature extraction of the time series, while the pooling layer condenses the extracted features using the maximum pooling method to generate crucial feature information.

The BiLSTM hidden layer model effectively learns the internal dynamic changes of the local features extracted by CNN and iteratively extracts intricate global features from the local features. The BiLSTM hidden layer generates features that are adeptly harnessed by the Attention mechanism, which discerns the significance of temporal information. This facilitates the extraction of profound temporal dependencies and enhances the utilization of the displacement time series' temporal characteristics. By preserving historical information and emphasizing critical historical time points, the Attention mechanism mitigates the impact of superfluous information on the displacement prediction outcomes. The outputs from the Attention layer serve as the input for the fully connected layer, which then precisely yields the final prediction of displacement. In optimizing the network parameters for this study, the Adam optimization algorithm is adopted to meticulously adjust the parameters across the layers, with the mean square error (MSE) serving as the loss function. The architecture of the combined CNN-BiLSTM-Attention model is depicted in Figure 4.

### 2.5.5 Displacement prediction processTo predict landslide displacement using the model, follow these steps with confidence

(1) The original landslide displacement time series is divided into three sub-sequences by using the SSA-VMD model. The landslide's cumulative displacement can be broken down into three sub-sequences: trend term displacement $T(t)$, periodic term displacement $P(t)$, and random term displacement $R(t)$, determined by the optimal fitness function value.

(2) The influence factor sequence is decomposed into two sub-sequences using SSA-VMD. The low-frequency and high-frequency parts of the influence factor sequence are represented by these sub-sequences. Using the optimal fitness function, we derived the optimal decomposition subsequence. We then calculated the maximum MIC and GRA values for the decomposition subsequence of each factor and displacement subsequence. Our comprehensive analysis allowed us to confidently assess the correlation between the influencing factor subsequence and the displacement subsequence.

(3) The input data is divided into a training dataset and a verification dataset based on the predetermined sequence of each displacement term and influencing factor. A single-factor CNN-BiLSTM-Attention model was constructed and trained for predicting trend term displacement, while a multi-factor CNN-BiLSTM-Attention model was established and trained for predicting periodic term displacement and random term displacement.

(4) Ultimately, the predicted values of trend displacement, periodic displacement, and random displacement are accumulated to form the cumulative landslide displacement prediction results, which are then compared with the cumulative landslide displacement monitoring results, and the predictive performance of the new model is evaluated. The process of the combined landslide displacement prediction model, involving SSA-VMD and CNN-BiLSTM-Attention, is confidently illustrated in Figure 5.

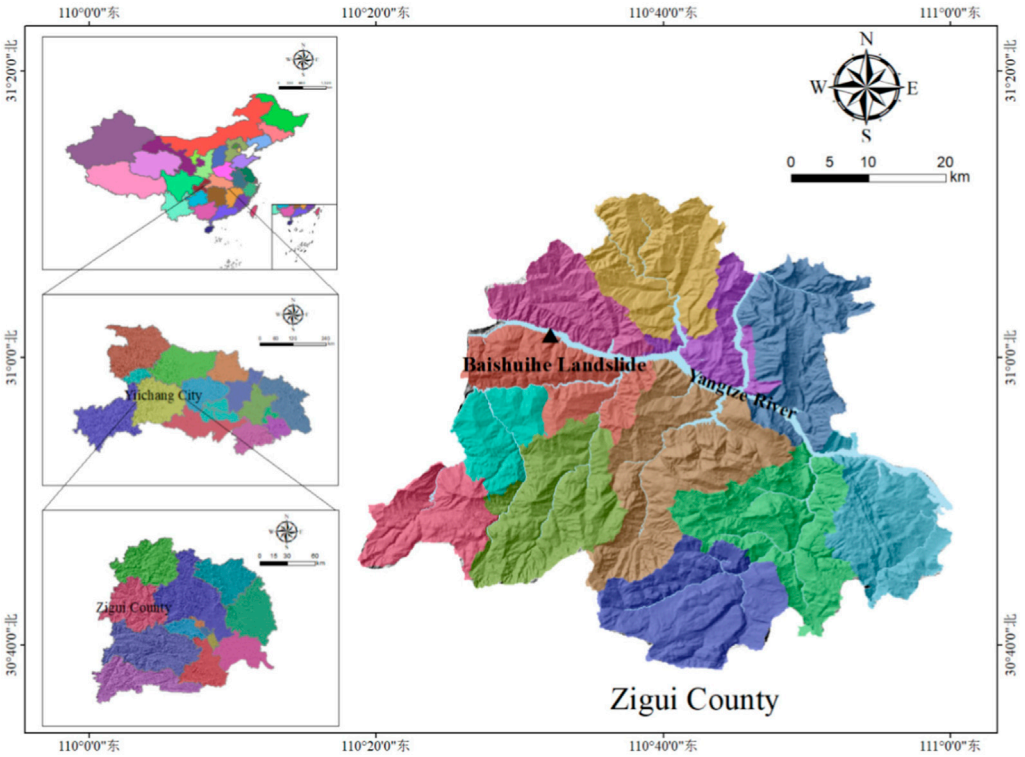



FIGURE 6
The geographical location of Baishuihe landslide.

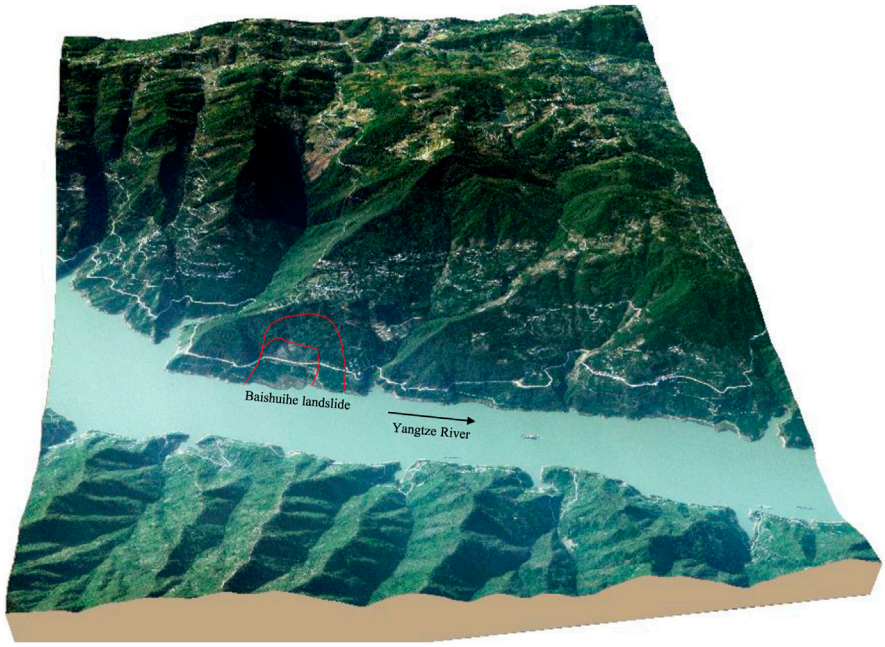



FIGURE 7
The topography of Baishuihe landslide.

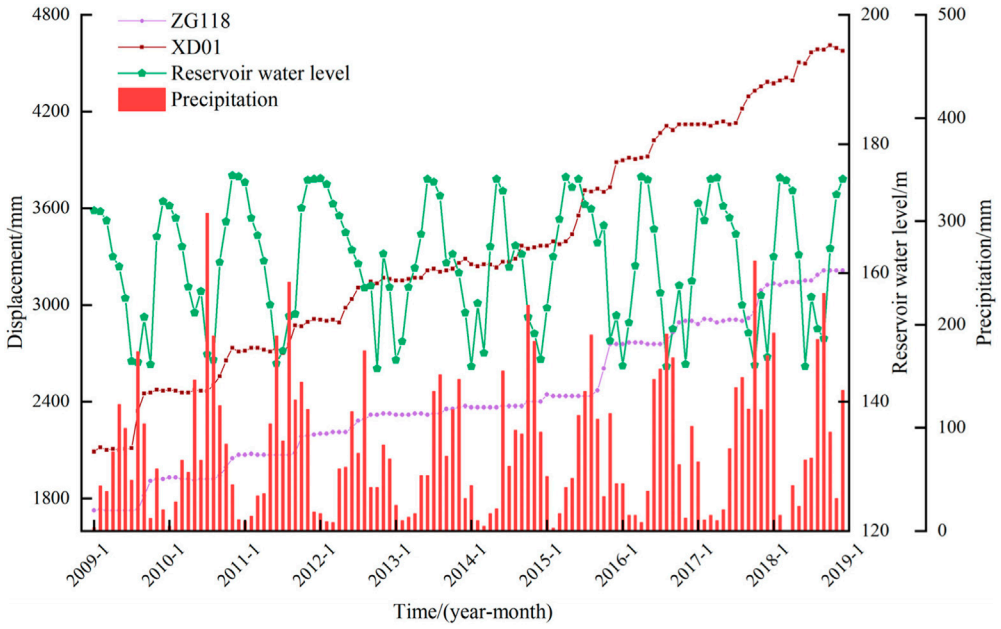



FIGURE 8
Displacement and environmental data variation in the Baishuihe landslide.

TABLE 1 SSA-VMD optimization results.

| Monitoring point | α | τ |
|---|---|---|
| ZG118 | 74.19 | 0.56 |
| XD01 | 90.13 | 0.17 |

# 3 Case study

## 3.1 Engineering geological survey and displacement analysis of Baishuihe landslide

The Baishuihe landslide, located in Zigui County within the Three Gorges Reservoir region, as illustrated in Figure 6 exhibits a monoclinic bedding slope structure, This structure is characterized by a gradient that is elevated in the south and decreases towards the north, aligning in a stepwise fashion towards the Yangtze River. The elevation measures approximately 410 m at the landslide's trailing edge and descends below the 135 m water level at its leading edge. The overall inclination of the Baishuihe landslide is estimated at 30°, with its topographical layout depicted in Figure 7. Since the commencement of monitoring activities in 2003, the landslide has experienced numerous significant deformation events. Geological surveys of the Baishuihe region elucidate the landslide's irregular 'U-shaped' configuration, extending 500 m in length from north to south, and 430 m across from east to west, covering an area of approximately $21.5 \times 10^4$ m$^2$. The sliding mass maintains an average thickness of about 30 m, culminating in a volume of roughly $645 \times 10^4$ m$^3$, with the principal direction of slide oriented at 20°.

Figure 8 demonstrates that every displacement change is linked to a rise in rainfall and a substantial shift in reservoir water level. Rainfall has an impact on the stability of the landslide by influencing the strength of rock and soil, physical and mechanical parameters, and pore water pressure [55–57]. The landslide motion state is influenced by the reservoir water level through hydrostatic pressure, hydrodynamic pressure, pore water pressure, and other factors [58,59]. The reservoir area experiences a flood season from May to September each year, resulting in increased rainfall and a wider fluctuation and influence range of the reservoir water level. Conversely, the non-flood season occurs from October to April of the following year, during which rainfall is scarce and the deformation of the landslide tends to be less severe. The periodic influence of reservoir water levels and rainfall causes the displacement of the landslide to exhibit a step-type characteristic.

## 3.2 Landslide displacement decomposition of VMD

The SSA utilizes a population size of 50 and a maximum of 100 iterations. , The optimization ranges for the penalty factor α and rise time step $\tau$ are [0.1,1000] and [0,1], respectively. Table 1 displays the optimization results, while Figures 9, 10 shows the corresponding decomposition results.

## 3.3 Selection of landslide displacement influencing factors

### 3.3.1 Analysis of landslide displacement influence factors

Examining the progression traits of the Baishuihe landslide, and building upon existing domestic and international research, many scholars have traditionally narrowed down the influencing factors of landslide displacement to rainfall and reservoir water level changes.

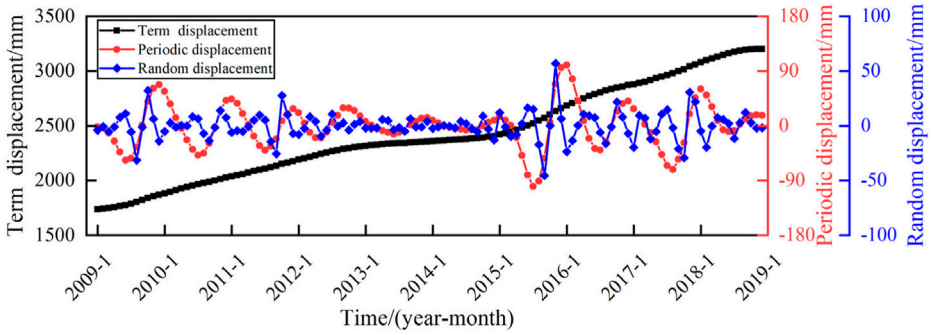



FIGURE 9
Decomposition results of ZG118 cumulative displacement.

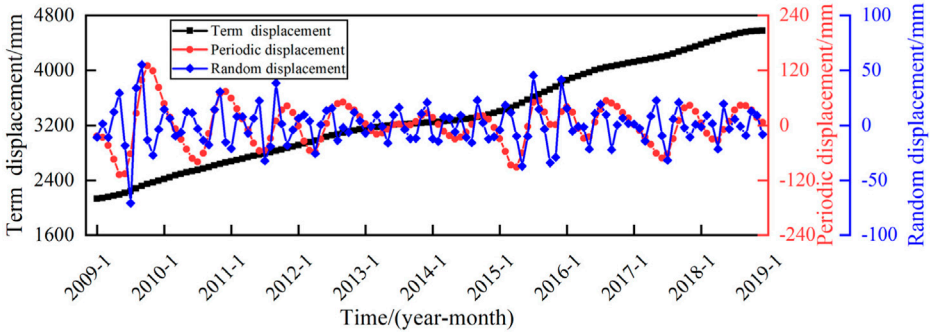



FIGURE 10
Decomposition results of XD01 cumulative displacement.

However, Figure 8 illustrates a discrepancy where the peak increase in landslide displacement occurred in 2015, despite there being neither the highest rainfall nor the most significant variation in reservoir water levels that year. This suggests that rainfall and reservoir water level changes do not singularly dictate landslide displacement. Such a dynamic may be attributed to the deformation evolution state of the landslide at the time, with different states responding variably to external influences. For instance, in phases where the landslide maintains relative stability, it is less likely to experience significant displacements, even when subjected to intense external forces. Conversely, in a state of instability, even moderate external influences can induce substantial movements [60,61]. Thus, it becomes evident that a landslide's deformation response hinges not only on the magnitude of the external triggers but also intimately connects with its current evolutionary stage. Accordingly, this study advances beyond conventional considerations by incorporating the displacement evolution state of the landslide as an additional input characteristic for the prediction model.

The analysis above identifies the indicators that have the most influence on landslide displacement. These are 1-month cumulative rainfall ($P_1$) and 2-month cumulative rainfall ($P_2$). Additionally, the monthly average reservoir water level elevation ($R_1$), the amplitude of reservoir water level in the previous month ($R_2$), and the amplitude of reservoir water level in the previous 2 months ($R_3$) are the influential factors for reservoir water level on landslide displacement. This information is presented with confidence and clarity to ensure a thorough understanding of the topic. To characterize the evolution state of landslide displacement, we choose the displacement ($S_1$) from the previous month and the displacement ($S_2$) from the previous 2 months.

The VMD algorithm decomposes the influencing factor sequence into high-frequency and low-frequency sequences. The high-frequency factors, such as $P_1^U$, $P_2^U$, $R_1^U$, $R_2^U$, $R_3^U$, $S_1^U$ and $S_2^U$, are used as the influencing factors of the random term displacement. The low-frequency factors, such as $P_1^L$, $P_2^L$, $R_1^L$, $R_2^L$, $R_3^L$, $S_1^L$ and $S_2^L$, are used as the influencing factors of the periodic term displacement. The correlation between the decomposition sequence of the impact factor and the periodic displacement and random term displacement sequence is comprehensively measured using the MIC and GRA.

### 3.3.2 Sequence decomposition of optimal VMD displacement influencing factors

The SSA utilizes a population size of 50 and a maximum number of iterations set to 100, with optimization ranges for the penalty factor α and rise time step τ being [0.01,1000] and [0,1], respectively. The optimization results are presented in Table 2, while the corresponding decomposition results are illustrated in Figures 11–14.

TABLE 2 SSA-VMD optimization results.

| Influencing factors | Precipitation | | Reservoir water level | | | The landslide state of ZG118 | | The landslide state of XD01 | |
|---|---|---|---|---|---|---|---|---|---|
| | $P_1$ | $P_2$ | $R_1$ | $R_2$ | $R_3$ | $S_1$ | $S_2$ | $S_1$ | $S_2$ |
| α | 0.65 | 0.15 | 0.19 | 30.16 | 20.20 | 12.18 | 1.15 | 1.19 | 12.17 |
| τ | 0.57 | 0.57 | 0.27 | 0.46 | 0.24 | 0.19 | 0.21 | 0.11 | 0.26 |

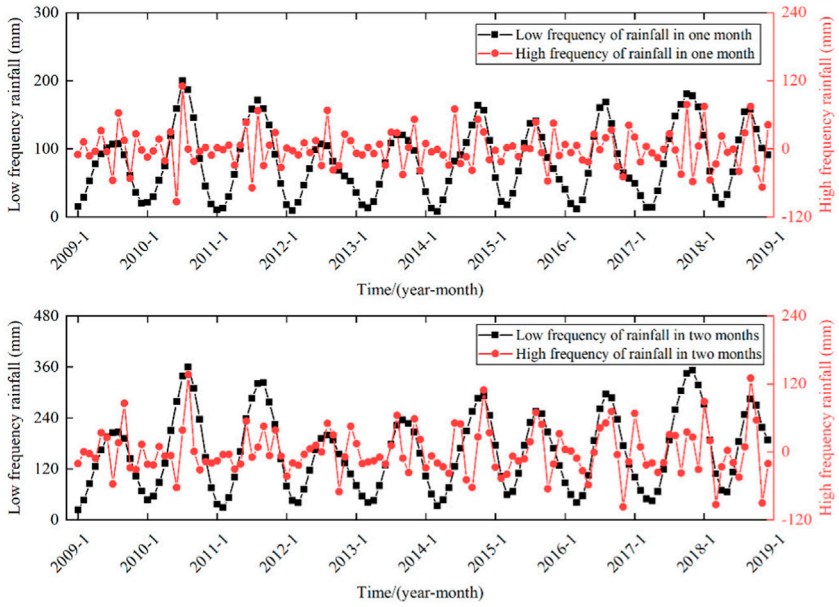



FIGURE 11
Decomposition results of cumulative rainfall.

### 3.3.3 Correlation analysis between displacement and influence factors

In the quest to elucidate the correlation between landslide displacement and its influencing factors, it is imperative to conduct a detailed analysis and decomposition of these factors. Selecting highly correlated factors is crucial for enhancing the predictive accuracy and efficacy of the model. Nonetheless, the availability of sufficiently high-quality data for model training is paramount. The inclusion of factors with minimal correlation risks incorporating extraneous data, potentially diminishing the precision and effectiveness of the landslide displacement prediction model. Optimally selected influencing factors can markedly elevate both the performance and accuracy of the model. In existing research, most scholars predominantly utilize a single method to assess the correlation between displacement components and influencing factors. However, a sole evaluation method can only provide perspective from a singular angle, resulting in a one-sided assessment and the loss of significant data portions. To address this, This study incorporates the MIC-GRA fusion method for a more comprehensive selection. Table 3 present the computation outcomes of both methods and, through comparative analysis in the subsequent prediction, the supremacy of this method is affirmed.

## 3.4 Displacement prediction results and analysis

### 3.4.1 Trend displacement prediction

The displacement of a landslide is influenced by topography, geological structure, and rock and soil properties. The displacement trend exhibits a monotonically increasing curve over time. While polynomial functions are frequently used in existing research to fit the trend displacement sequence, it may be necessary to perform piecewise fitting due to differences in deformation characteristics across different stages. This is because a single function often fails to fit the entire trend displacement curve. This paper presents a single-factor CNN-BiLSTM-Attention model for predicting trend item displacement. The model takes the displacement values of the previous month, the first 2 months, the first 3 months, the displacement change value of the previous month and the change value of the previous 2 months as input. The prediction results are presented in Figure 15 which show that monitoring points ZG118 and XD01 have $R^2$ values of 0.995 and 0.999, respectively, with corresponding RMSE values of 3.195 and 6.573.

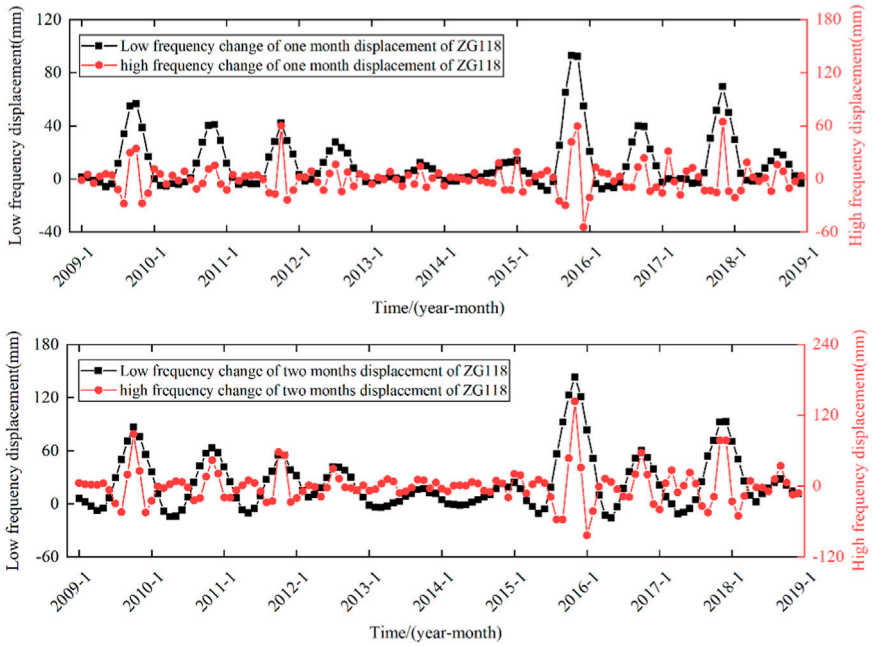



FIGURE 12
Decomposition results of displacement variation of ZG118.

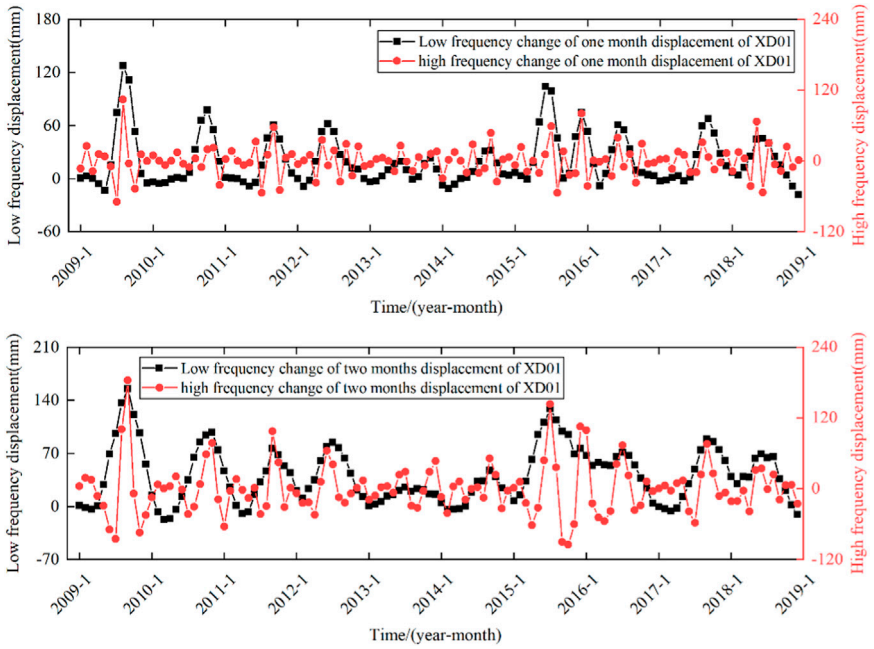



FIGURE 13
Decomposition results of displacement variation of XD01.

### 3.4.2 Period displacement prediction

In this paper, the factor sequence was selected based on a MIC value greater than 0.25 and a GRA value greater than 0.60. We conducted multiple selections and trial calculations to ensure complete in our final selection. The periodic term displacement sequence and the low-frequency influencing factor sequence were chosen and will be used as input for the prediction model. A multi-factor CNN-BiLSTM-Attention model was constructed for training and prediction. The predictive outputs for this model are showcased in Figure 16, which show that monitoring points ZG118 and XD01 have $R^2$ values of 0.994 and 0.995, respectively, with corresponding RMSE values of 1.670 and 1.798.

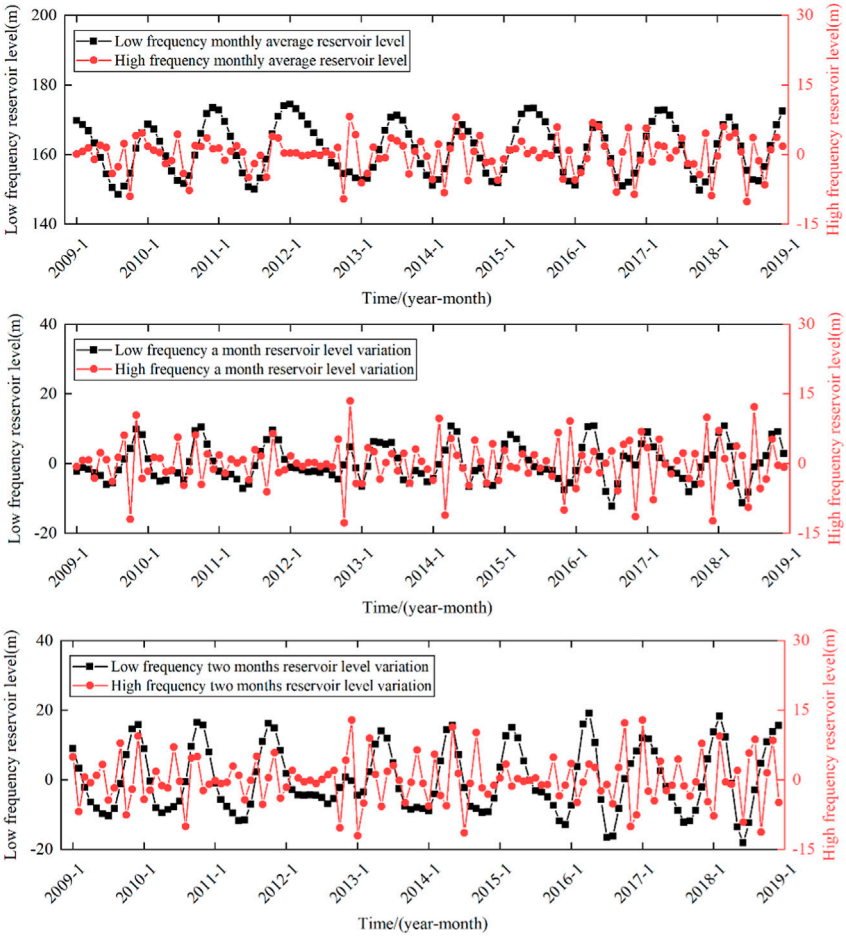



FIGURE 14
Decomposition results of reservoir level.

TABLE 3 Correlation coefficient between periodic term displacement and influencing factors.

| Influencing factors | | Periodic displacement | | | | Random displacement | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | XD01 | | ZG118 | | XD01 | | ZG118 | |
| | | MIC | GRA | MIC | GRA | MIC | GRA | MIC | GRA |
| rainfall | $P_1^L$ | 0.255 | 0.67 | 0.403 | 0.63 | 0.252 | 0.66 | 0.175 | 0.78 |
| | $P_2^L$ | 0.308 | 0.68 | 0.347 | 0.65 | 0.316 | 0.64 | 0.322 | 0.75 |
| Reservoir water level | $R_1^L$ | 0.234 | 0.63 | 0.307 | 0.66 | 0.260 | 0.69 | 0.290 | 0.71 |
| | $R_2^L$ | 0.287 | 0.69 | 0.220 | 0.73 | 0.253 | 0.67 | 0.204 | 0.77 |
| | $R_3^L$ | 0.373 | 0.67 | 0.303 | 0.72 | 0.192 | 0.66 | 0.253 | 0.75 |
| State of landslide | $S_1^L$ | 0.289 | 0.66 | 0.299 | 0.65 | 0.563 | 0.65 | 0.421 | 0.87 |
| | $S_2^L$ | 0.393 | 0.70 | 0.369 | 0.68 | 0.386 | 0.60 | 0.574 | 0.85 |

### 3.4.3 Random term displacement prediction

In this study, factors with a MIC value exceeding 0.25 and a GRA value above 0.60 were chosen from the random term displacement sequence and the high-frequency influencing factor sequence. These factors served as inputs for the multi-factor CNN-BiLSTM-Attention model, which was utilized for training and forecasting. The predictive

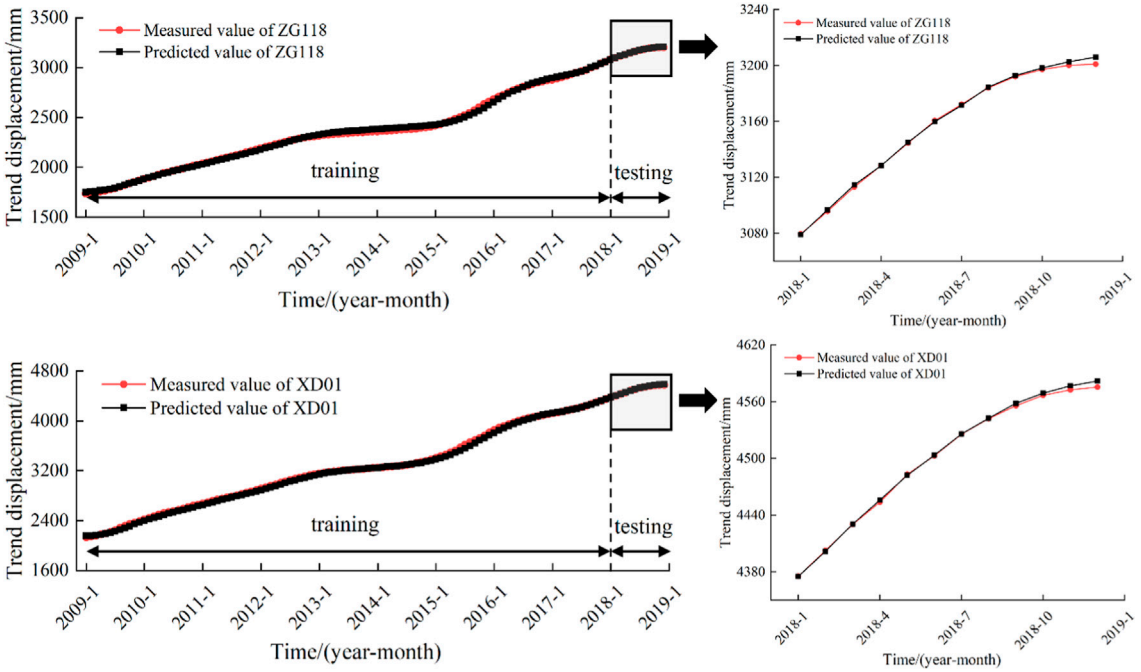



FIGURE 15
Displacement prediction results of trend items.

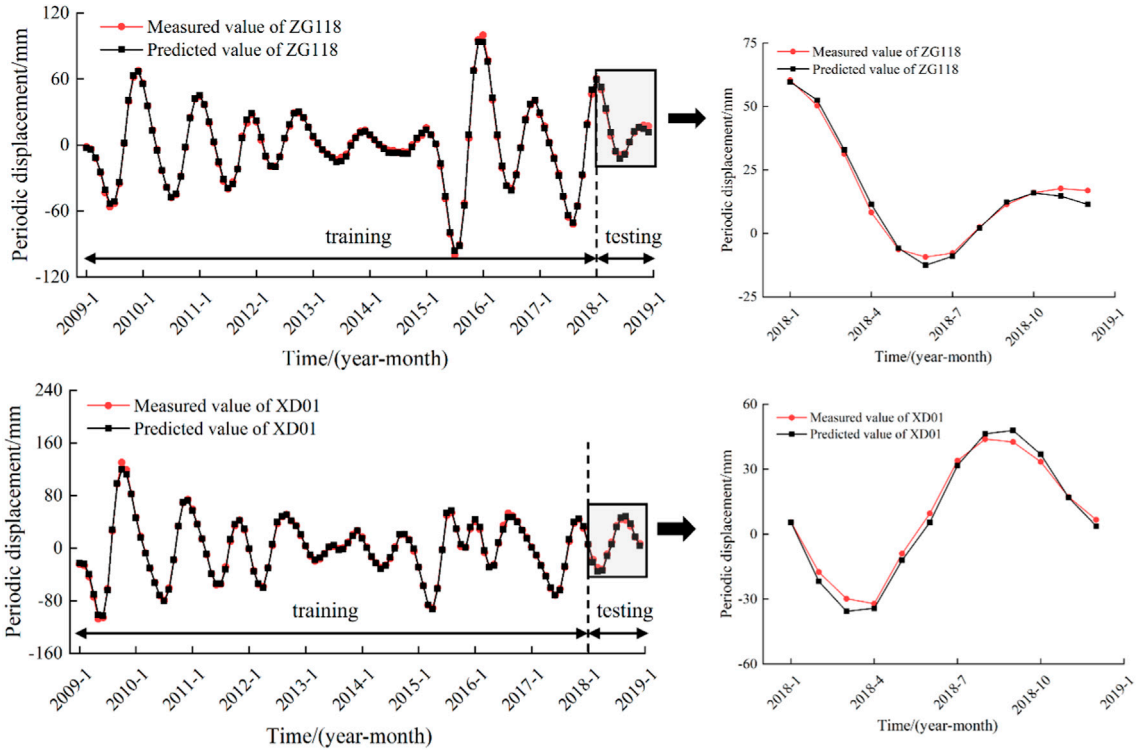



FIGURE 16
Prediction results of periodic term displacement.

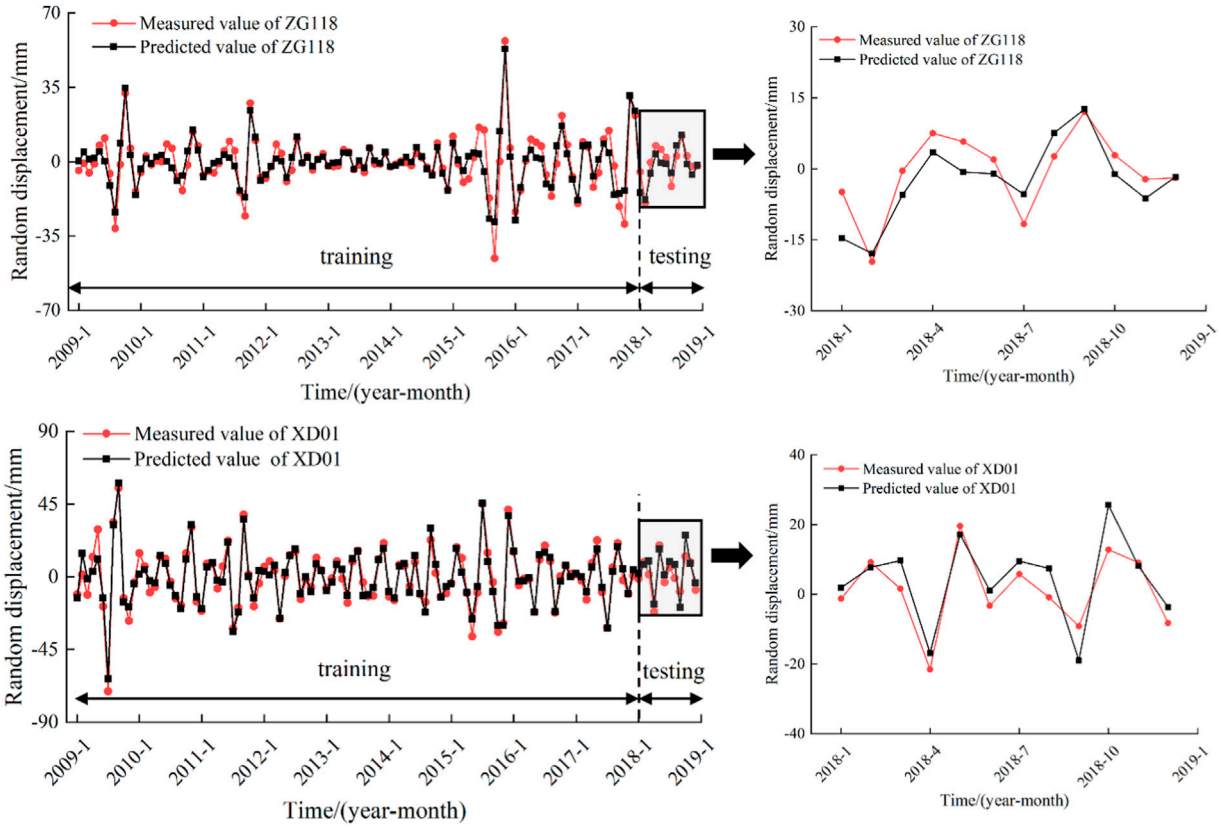



FIGURE 17
Displacement prediction results of random terms.

outputs for this model are showcased in Figure 17; they demonstrate an $R^2$ value of 0.723 and an RMSE value of 4.296 at monitoring point ZG118, alongside an $R^2$ value of 0.612 and an RMSE value of 5.472 at monitoring point XD01.

### 3.4.4 Cumulative displacement prediction

By summing the prediction outcomes of trend term displacement, periodic term displacement, and random term displacement in accordance with time series summation principles, cumulative predictions for landslide displacement are derived. These results are illustrated in Figure 18, exhibiting $R^2$ values of 0.975 for monitoring point ZG118 and 0.988 for XD01. Correspondingly, the RMSE values are reported as 12.458 mm for ZG118 and 9.579 mm for XD01. Such high $R^2$ values alongside low RMSE values attest to the model's robust prediction accuracy, thereby reaffirming its efficacy in forecasting landslide events.

## 3.5 Comparative analysis

### 3.5.1 Selection of impact factors

To enhance the predictive performance, this research adopts several models CNN-BiLSTM-Attention, GRA-CNN-BiLSTM-Attention, MIC-CNN-BiLSTM-Attention, and (MIC- GRA)-CNN-BiLSTM-Attention for the prediction and comparative analysis of the two components of landslide displacement under uniform conditions. The prediction results of various influencing factor selection methods are shown in Table 4.

From Table 4, it can be deduced that using the GRA algorithm or MIC algorithm effectively selects influencing factors. The predictive results indicate that the models combined with these two algorithms exhibit higher precision, which reflects the role of both algorithms in selecting influencing factors. Moreover, the model that utilizes the MIC- GRA evaluation method to select related influencing factors achieves the highest precision in prediction results, indirectly showcasing the superiority of the MIC- GRA algorithm. This is because when the MIC- GRA algorithm is integrated with the model, it can select influencing factors from two different perspectives, eliminating data with low relevance and retaining high-relevance influencing factors. Due to the input of effective influencing factors, the predictive accuracy of the model combined with the MIC- GRA algorithm is enhanced.

### 3.5.2 Comparative analysis of periodic displacement prediction

The predictive results of the CNN-BiLSTM-Attention model were compared with the static machine learning models such as the BP Neural Network and SVM models, and the deep learning models'

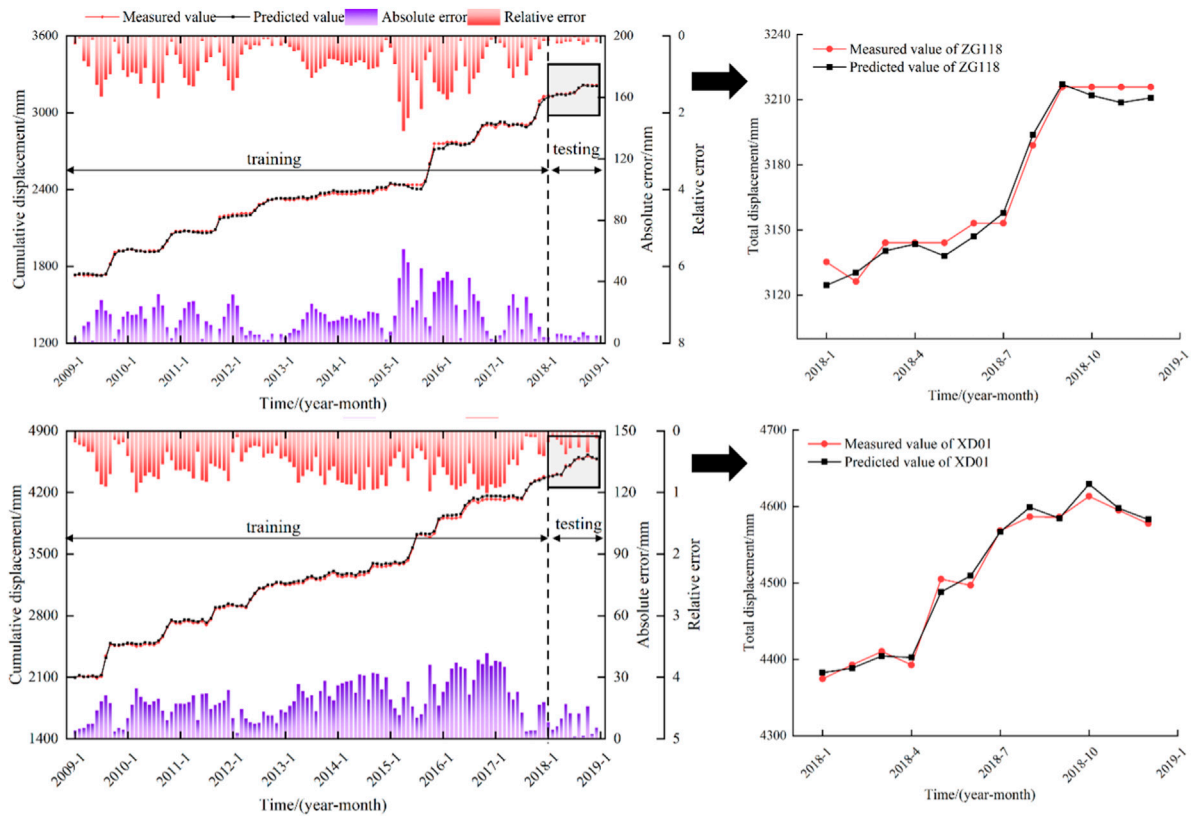



FIGURE 18
Cumulative displacement prediction results.

TABLE 4 Comparison of prediction performances of the CNN-BiLSTM-Attention model under different inputs.

| Models | Periodic term displacement | | | | Random term displacement | | | |
|---|---|---|---|---|---|---|---|---|
| | ZG118 | | XD01 | | ZG118 | | XD01 | |
| | RMSE/mm | $R^2$ | RMSE/mm | $R^2$ | RMSE/mm | $R^2$ | RMSE/mm | $R^2$ |
| CNN-BiLSTM-Attention | 6.517 | 0.901 | 7.024 | 0.911 | 7.015 | 0.483 | 9.561 | 0.323 |
| GRA-CNN-BiLSTM-Attention | 5.646 | 0.928 | 6.854 | 0.930 | 5.258 | 0.585 | 8.404 | 0.332 |
| MIC-CNN-BiLSTM-Attention | 5.036 | 0.943 | 6.854 | 0.930 | 4.478 | 0.699 | 8.063 | 0.385 |
| (MIC-GRA)-CNN-BiLSTM-Attention | 1.670 | 0.994 | 1.798 | 0.995 | 4.296 | 0.723 | 5.472 | 0.612 |

predictions such as LSTM, BiLSTM, CNN-BiLSTM, all of which are widely used in landslide displacement prediction. The predictive results of each model are presented in Table 5.

Table 5 details the predictive outcomes, illustrating that the CNN-BiLSTM-Attention model achieves superior accuracy in forecasting periodic displacement when contrasted with the standalone BP and SVM models, which are inherently static. This improvement is attributed to the dynamic features of the BiLSTM model, which is adept at processing the dynamic nature of landslide displacement sequences via its bidirectional training capability. Concurrently, the convolutional neural networks and attention mechanisms facilitate the distillation of pertinent information, simplifying data complexity and thus enhancing accuracy for periodic terms. This conclusion is further reinforced through the comparative analysis with the LSTM, BiLSTM, and CNN-BiLSTM models. Collectively, the evidence indicates that the prediction accuracy of deep learning models eclipses that of traditional machine learning models, and that combined models deliver improved results over singular models.

### 3.5.3 Random term displacement prediction model comparative analysis

The predictive efficacy of the CNN-BiLSTM-Attention model in forecasting landslide displacement is assessed by comparing it with

TABLE 5 Comparison of periodic term displacement prediction models.

| Models | ZG118 | | XD01 | |
|---|---|---|---|---|
| | RMSE/mm | $R^2$ | RMSE/mm | $R^2$ |
| (MIC-GRA)-BP | 11.476 | 0.796 | 10.932 | 0.848 |
| (MIC-GRA)-SVM | 12.170 | 0.695 | 11.910 | 0.786 |
| (MIC-GRA)-LSTM | 5.341 | 0.931 | 6.862 | 0.915 |
| (MIC-GRA)-BiLSTM | 3.913 | 0.961 | 3.717 | 0.967 |
| (MIC-GRA)-CNN-BiLSTM | 3.003 | 0.972 | 2.912 | 0.975 |
| (MIC-GRA)-CNN-BiLSTM-Attention | 1.670 | 0.994 | 1.798 | 0.995 |

TABLE 6 Comparison of random item displacement prediction models.

| Models | ZG118 | | XD01 | |
|---|---|---|---|---|
| | RMSE/mm | $R^2$ | RMSE/mm | $R^2$ |
| (MIC-GRA)-LSTM | 8.506 | 0.309 | 13.587 | 0.252 |
| (MIC-GRA)-BiLSTM | 8.331 | 0.357 | 12.835 | 0.267 |
| (MIC-GRA)-CNN-BiLSTM | 8.054 | 0.448 | 10.212 | 0.312 |
| (MIC-GRA)-CNN-BiLSTM-Attention | 4.296 | 0.723 | 5.472 | 0.612 |

the LSTM, BiLSTM, and CNN-BiLSTM models. Results of this comparative analysis are delineated in Table 6.

Table 6 illustrates the superior prediction accuracy of the CNN-BiLSTM-Attention model compared to its LSTM, BiLSTM, and CNN-BiLSTM counterparts for random item displacement. This heightened accuracy is ascribed to the model's robust handling of the random item displacement sequence, characterized by its high frequency and considerable volatility. The CNN-BiLSTM-Attention model excels beyond traditional LSTM and BiLSTM models, particularly in capturing nonlinear information embedded within time series data. This model adeptly retains vital information by employing the bidirectional training capabilities of BiLSTM. In addition, the attention mechanism's capacity to assign differentiated weights to disparate data points streamlines the process, culminating in the effective and precise training of random item displacement sequences.

## 4 Discussion

Accurately assessing reservoir landslide deformation is vital for averting landslide calamities, given the considerable nonlinearity and intricacy inherent in landslide displacement and its causative factors. This study introduces a data-driven framework comprising a deep learning ensemble model twinned with an optimal variational mode decomposition, designed to forecast future landslide movements. This framework's benefits are twofold. First, it applies the SSA-VMD algorithm to decompose the landslide displacement sequence and its influencing factors, thereby improving the time series displacement prediction model's efficacy. Second, this trailblazing research harnesses a CNN-BiLSTM-attention ensemble model to anticipate reservoir landslide shifts. This deep learning ensemble model synergizes the strengths of individual models, providing enhanced capability in feature extraction from datasets marked by nonlinearity and complexity.

While various displacement decomposition methods offer substantial decomposition effects, it is essential to highlight that the SSA-VMD model introduced in this study distinguishes itself by its ability to accurately capture random term displacement. Nevertheless, the current limitation in making more precise predictions stems from the inadequate availability of monitoring data on relevant influencing factors.

Moreover, existing landslide displacement monitoring data are exclusively sourced from slopes already exhibiting deformation. The inherent nonlinearity of slope characteristics complicates the task of forecasting landslide deformation accurately using historical, static data. Future research endeavors must focus on incorporating real-time monitoring data into predictive models. Such integration would not only enhance the precision and promptness of the models' predictions but also render them more effective for early warning systems.

Prediction methods based on single-point displacement remain central within the domain of landslide deformation research. However, the inherent uncertainty in landslide systems makes some degree of error in traditional point prediction methods inevitable. To address this, our study applies prediction intervals to improve the accuracy of landslide displacement forecasts [62]. Although the current focus is primarily on reservoir landslides influenced by hydrological factors, the scope of the predictive model should be expanded. Future developments could include additional influencing factors such as soil mechanics and seismic

activity, paving the way for a more generalized displacement prediction model.

## 5 Conclusion

In this study, we introduce the (SSA-VMD)-(CNN-BiLSTM-Attention) model for predicting landslide displacement, which synergizes the SSA-VMD technique with the CNN-BiLSTM-Attention model applied to landslide displacement sequences and their influencing factors. Employed in the study of Baishuihe landslide's displacement prediction, the research leads to the following conclusion:

(1) In the VMD model, the SSA algorithm is utilized to dynamically optimize parameters, reducing the influence of subjective assumptions and avoiding the laborious process of manual parameter tuning. When designing the innovative fitness functions, the reliability and decomposition efficiency of the VMD model are enhanced by adopting sample entropy and root mean square error.

(2) The SSA-VMD algorithm allows for the extraction of subsequences of landslide displacement and subsequences of influencing factors, enabling an in-depth analysis of the relationships between landslide displacement, rainfall, reservoir water levels, and the state of landslide displacement. The correlations between the displacement subsequences and influencing factors are calculated using the MIC and GRA methods. Furthermore, the integration of MIC-GRA as a fusion technique is utilized for selecting significant influencing factors for landslide displacement. The results indicate that using influencing factors selected by the MIC-GRA method as input data can significantly enhance prediction accuracy, demonstrating that this method can improve the effectiveness and efficiency of the input data. By eliminating less relevant data, the predictive accuracy of the model is increased.

(3) The study introduces a novel integrated model, CNN-BiLSTM-Attention, designed for training and predicting landslide displacement. This composite model combines the strengths of CNN, BiLSTM, and the Attention mechanism to adeptly extract essential information from landslide displacement data. The CNN component handles feature extraction, while BiLSTM processes both past and future data, and the Attention mechanism assigns variable weights to the data, thereby optimizing the prediction process for landslide displacement. Empirical results suggest that the proposed model surpasses both single and dual combined models in prediction accuracy. The pronounced accuracy of this model better captures the step process of landslides and serves as a foundational study for the prediction and early warning of similar landslide events.

## Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

## Author contributions

RW: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Writing–original draft, Writing–review and editing. YL: Data curation, Formal Analysis, Investigation, Methodology, Supervision, Writing–review and editing. YY: Data curation, Formal Analysis, Methodology, Writing–review and editing. WX: Conceptualization, Funding acquisition, Methodology, Writing–review and editing. YW: Data curation, Methodology, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work is supported by the National Natural Science Foundation of China (No. 51939004).

## Acknowledgments

We thank the National Field Observation and Research Station of Landslides in the TGRA of Yangtze River for their help in providing monitoring data for this study.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note



## References

1. Peng L, Niu RQ, Wu T. Time series analysis and support vector machine for landslide displacement prediction. *J Zhejiang University(Engineering Science)* (2013) 47(09):1672–9. doi:10.3785/j.issn.1008-973X.2013.09.024

2. Xu Q, Huang RQ, Li XZ. Research progress in time forecast and prediction on of landslides. *Adv Earth Sci* (2004) 19(03):478–83. doi:10.11867/j.issn.1001-8166.2004.03.0478

3. Li Hj., He YS, Xu Q, Deng JH, Li WL, Wei Y, et al. Sematic segmentation of loess landslides with STAPLE mask and fully connected conditional random field. *Landslides* (2023) 20(2):367–80. doi:10.1007/s10346-022-01983-8

4. Xu F, Wang Y, Du J, Ye J. Study of displacement prediction model of landslide based on time series analysis. *Chin J Rock Mech Eng* (2011) 30(04):746–51.

5. Zhang J, Yin KL, Wang JJ, Huang MF. Displacement prediction of Baishuihe landslide based on time series and PSO-SVR model. *Chin J Rock Mech Eng* (2015) 34(02):382–91. doi:10.13722/j.cnki.jrme.2015.02.017

6. Zhou C, Yin KL, Cao Y, Ahmed B. Application of time series analysis and PSO-SVM model in predicting the Bazimen landslide in the Three Gorges Reservoir. *Eng Geology* (2016) 204:108–20. doi:10.1016/j.enggeo.2016.02.009

7. Liu YL, Yin KL, Wang Y, Wang W. Study of landslide deformation prediction based on EMD and neural network. *Saf Environ Eng* (2013) 20(04):14–7.

8. Xu SL, Niu RQ. Displacement prediction of Baijiabao landslide based on empirical mode decomposition and long short term memory neural network in Three Gorges area, China. *Comput Geosci* (2018) 111:87–96. doi:10.1016/j.cageo.2017.10.013

9. Zhang K, Zhang K, Bao R, Liu XH, Qi FF. Intelligent prediction of landslide displacements based on optimized empirical mode decomposition and K-Mean clustering. *Rock Soil Mech* (2021) 42(01):211–23. doi:10.16285/j.rsm.2020.1300

10. Deng DM, Liang Y. Displacement prediction method based on ensemble empirical mode decomposition and support vector machine regression— a case of landslides in Three Gorges Reservoir area. *Rock Soil Mech* (2017) 38(12):3660–9. doi:10.16285/j.rsm.2017.12.034

11. Wang ZH, Nie W, Xu HH, Jian WB. Prediction of landslide displacement based on EEMD-Prophet-LSTM. *J Univ Chin Acad Sci* (2023) 40(04):514–22. doi:10.7523/j.ucas.2022.002

12. Du H, Song DQ, Chen Z, Shu HP, Guo ZZ. Prediction model oriented for landslide displacement with step-like curve by applying ensemble empirical mode decomposition and the PSO-ELM method. *J Clean Prod* (2020) 270:122248. doi:10.1016/j.jclepro.2020.122248

13. Zhang KX, Niu RQ, Hu YJ, Wu XL. Landslide displacement prediction based on wavelet transform and external cause. *J China Univ Mining& Tech* (2017) 46(04):924–31. doi:10.13247/j.cnki.jcumt.000716

14. Zhou C, Yin K, Cao Y, Intrieri E, Ahmed B, Catani F. Displacement prediction of step-like landslide by applying a novel kernel extreme learning machine method. *Landslides* (2018) 15:2211–25. doi:10.1007/s10346-018-1022-0

15. Zhou C, Yin KL, Huang FM. Application of the chaotic sequence WA-ELM coupling model in landslide displacement prediction. *Rock Soil Mech* (2015) 36(09):2674–80. doi:10.16285/j.rsm.2015.09.030

16. Luo HY, Jang YN, Xu Q, Tang B. Displacement prediction of reservoir bank landslide based on optimal decomposition mode and GRU model. *Geomatics Inf Sci Wuhan Univ* (2023) 48(05):702–9. doi:10.13203/j.whugis20200610

17. Jiang YH, Wang W, Zhou LF, Wang RB, Liu SP. Research on dynamic prediction model of landslide displacement based on particle swarm optimization-variational mode decomposition, nonlinear autoregressive neural network with exogenous inputs and gated recurrent unit. *Rock Soil Mech* (2022) 43(S1):601–12. doi:10.16285/j.rsm.2021.0247

18. Xing Y, Yue JP, Chen C, Qin YL, Hu J. A hybrid prediction model of landslide displacement with risk-averse adaptation. *Comput Geosci* (2020) 141:104527. doi:10.1016/j.cageo.2020.104527

19. Li LW, Wu YP, Miao FS, Liao K, Zhang LF. Displacement prediction of landslides based on variational mode decomposition and GWO-MIC-SVR model. *Chin J Rock Mech Eng* (2018) 37(06):1395–406. doi:10.13722/j.cnki.jrme.2017.1508

20. Xu F, Fan CJ, Xu XJ, Li L, Ni JJ. Displacement prediction of landslide based on variational mode decomposition and AMPSO-SVM coupling model. *J Shanghai Jiaotong Univ* (2018) 52(10):1388–95+1416. doi:10.16183/j.cnki.jsjtu.2018.10.030

21. Zhou C, Yin KL, Cao Y, Huang FM. Displacement Prediction of step-like Landslide Based on the response of inducing factors and support vector machine. *Chin J Rock Mech Eng* (2015) 34(S2):4132–9. doi:10.13722/j.cnki.jrme.2014.0290

22. Yang F, Xu Q, Fan XM, Ye W. Prediction of landslide displacement time series based on support vector regression machine with artificial bee colony algorithm. *J Eng Geology* (2019) 27(04):880–9. doi:10.13544/j.cnki.jeg.2017-256

23. Du J, Yin KL, Chai B. Study of displacement prediction model of landslide based on response analysis of inducing factors. *Chin J Rock Mech Eng* (2009) 28(09):1783–9.

24. Wang RB, Zhang K, Wang W, Meng YD, Yang LL, Huan HF, Hydrodynamic landslide displacement prediction using combined extreme learning machine and random search support vector regression model. *Eur J Environ Civ Eng*, 27, 2020, 2345–57. doi:10.1080/19648189.2020.1754298

25. Li Hj., Xu Q, He YS, Deng JH. Prediction of landslide displacement with an ensemble-based extreme learning machine and copula models. *Landslides* (2018) 15:2047–59. doi:10.1007/s10346-018-1020-2

26. Wang Yk., Tang HM, Huang JS, Wen T, Ma JW, Zhang JR. A comparative study of different machine learning methods for reservoir landslide displacement prediction. *Eng Geology* (2022) 298:106544. doi:10.1016/j.enggeo.2022.106544

27. Yao W, Lian C, Cheng L. A dynamic probabilistic model for landslide displacement prediction. *Hydrogeology Eng Geology* (2015) 42(05):134–9+148. doi:10.16030/j.cnki.issn.1000-3665.2015.05.22

28. Yao W, Lian C. Prediction of landslide displacement based on reservoir computing and fractal interpolation. *J Yangtze River Scientific Res Inst* (2014) 31(12):43–8. doi:10.3969/j.issn.1001-5485.2014.12.009

29. Yang BB, Yin KL, Du J. A model for predicting landslide displacement based on time series and long and short term memory neural network. *Chin J Rock Mech Eng* (2018) 37(10):2334–43. doi:10.13722/j.cnki.jrme.2018.0468

30. Xing Y, Yue JP, Chen C. Interval estimation of landslide displacement prediction based on time series decomposition and long short-term memory network. *IEEE Access* (2020) 8:3187–96. doi:10.1109/access.2019.2961295

31. Yang BB, Yin KL, Lacasse S, Liu ZQ. Time series analysis and long short-term memory neural network to predict landslide displacement. *Landslides* (2019) 16:677–94. doi:10.1007/s10346-018-01127-x

32. Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. In: *31st international conference on machine learning*. Beijing, China: W&CP (2014). p. 1764–72.

33. Lin Z, Ji Y, Liang W, Sun X. Landslide displacement prediction based on time-frequency analysis and LMD-BiLSTM model. *Mathematics* (2022) 10(13):2203. doi:10.3390/math10132203

34. Zhang K, Zhang K, Cai C, Liu W, Xie J. Displacement prediction of step-like landslides based on feature optimization and VMD-Bi-LSTM: a case study of the Bazimen and Baishuihe landslides in the Three Gorges, China. *Bull Eng Geology Environ* (2021) 80:8481–502. doi:10.1007/s10064-021-02454-5

35. Niu Q, Cao AM, Chen XY, Zhou D. Short-term load forecasting based on flower pollination algorithm and BP neural network. *Power Syst Clean Energ* (2020) 36(10):28–32.

36. Liu D, Wei X, Wang WQ, Ye JH, Reng J. Short-term wind power prediction based on SSA-ELM. *Smart Power* (2021) 49(06):53–9+123.

37. Nava L, Carraro E, Reyes-Carmona C, Puliero S, Bhuyan K, Rosi A, et al. Landslide displacement forecasting using deep learning and monitoring data across selected sites. *Landslides* (2023) 20(10):2111–29. doi:10.1007/s10346-023-02104-9

38. Lin Z, Ji Y, Sun X. Landslide displacement prediction based on CEEMDAN method and CNN–BiLSTM model. *Sustainability* (2023) 15(13):10071. doi:10.3390/su151310071

39. Wang CY, Li LM, Wen ZZ, Zhang MY, Wei XW. Dynamic prediction of landslide displacement based on time series and CNN-LSTM. *Foreign Electron Meas Tech* (2022) 41(03):1–8. doi:10.19652/j.cnki.femt.2103321

40. Zhu ZL, Rao Y, Wu Y, Qi JN, Zhang Y. Research progress of attention mechanism in deep learning. *J Chin Inf Process* (2019) 33(06):1–11.

41. Tang FF, Tang TJ, Zhu HZ, Hu C, Ma Y, Li X. Rainfall landslide deformation prediction based on attention mechanism and Bi-LSTM. *Bull Surv Mapp* (2022)(09)74–9+104. doi:10.13474/j.cnki.11-2246.2022.0267

42. Dragomiretskiy K, Zosso D. Variational mode decomposition. *IEEE Transactions Signal Processing* (2013) 62(3):531–44. doi:10.1109/tsp.2013.2288675

43. Richman JS, Moorman JR. Physiological time-series analysis using approximate entropy and sample entropy. *Am J Physiology-Heart Circulatory Physiol* (2000) 278(6):H2039–49. doi:10.1152/ajpheart.2000.278.6.h2039

44. Xue JX *Research and application of A novel swarm intelligence optimization technique: sparrow search algorithm*. China: Donghua University (2021).

45. Kraskov A, Stögbauer H, Grassberger P. Estimating mutual information. *Phys Rev E* (2004) 69(6):066138. doi:10.1103/physreve.69.066138

46. Reshef DN, Reshef YA, Finucane HK, Grossman SR, Mcvean G, Turnbaugh PJ, et al. Detecting novel associations in large data sets. *Science* (2011) 334(6062):1518–24. doi:10.1126/science.1205438

47. Guo Z, Yu B, Hao M, Wang W, Jiang Y, Zong F. A novel hybrid method for flight departure delay prediction using Random Forest Regression and Maximal Information Coefficient. *Aerospace Sci Tech* (2021) 116:106822. doi:10.1016/j.ast.2021.106822

48. Huang X, Luo YP, Xia L. An efficient wavelength selection method based on the maximal information coefficient for multivariate spectral calibration. *Chemom Intell Lab Syst* (2019) 194:103872. doi:10.1016/j.chemolab.2019.103872

49. Zhou FY, Jin LP, Dong J. Review of convolutional neural network. *Chin J Comput* (2017) 40(06):1229–51.

50. Khan S, Rahmani H, Shah SAA, Bennamoun M *A guide to convolutional neural networks for computer vision. San Rafael*. USA: Morgan and Claypool Publishers (2018).

51. Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. In: Proceedings of International Conference on Acoustics, Speech and Signal

Processing Acoustics; 4-10, June 2023; Vancouver, Canada, 6. IEEE (2013). p. 645–6 649.

52. Khan S, Fazil M, Sejwal VK, Alshara MA, Alotaibi RM, Kamal A, et al. BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection. *J King Saud University-Computer Inf Sci* (2022) 34(7):4335–44. doi:10.1016/j.jksuci.2022.05.006

53. Ren JJ, Wei HH, Zou ZL, Hou TT, Yuan YL, Shen JQ, et al. Ultra-short-term power load forecasting based on CNN-BiLSTM-Attention. *Power Syst Prot Control* (2022) 50(08):108–16. doi:10.19783/j.cnki.pspc.211187

54. Knudsen EI. Fundamental components of attention. *Annu Rev Neurosci* (2007) 30:57–78. doi:10.1146/annurev.neuro.30.051606.094256

55. Zhu YJ, He N, Zhong W, Kong JM. Physical simulation study of deformation and failure accumulation layer slope caused by intermittent rainfall. *Rock Soil Mech* (2020) 41(12):4035–44. doi:10.16285/j.rsm.2020.0318

56. He KQ, Guo L, Chen WG. Research on displacement dynamic evaluation and forecast model of colluvial landslide induced by rainfall. *Chin J Rock Mech Eng* (2015) 34(S2):4204–15. doi:10.13722/j.cnki.jrme.2014.1010

57. Wang RB, Xia R, Xu WY, Wang HL, Qi J. Study on physical simulation of rainfall infiltration process of landslide accumulation body. *Adv Eng Sci* (2019) 51(04):47–54. doi:10.15961/j.jsuese.201900295

58. Liu XX, Xia YY, Zhang XS, Guo RQ. Effects of drawdown of reservoir water level on landslide stability. *Chin J Rock Mech Eng* (2005) 24(8):1439–44.

59. Tan LY, Huang RQ, Pei XJ. Deformation characteristics and inducing mechanisms of a super-large bedding rock landslide triggered by reservoir water level decline in Three Gorges Reservoir area. *Chin J Rock Mech Eng* (2021) 40(02):302–14. doi:10.13722/j.cnki.jrme.2020.0728

60. Liu Z, Guo D, Lacasse S, Li J, Yang B, Choi J. Algorithms for intelligent prediction of landslide displacements. *J Zhejiang University-SCIENCE A* (2020) 21(6):412–29. doi:10.1631/jzus.a2000005

61. Liu Y, Xu C, Huang B, Ren X, Liu C, Chen BHandZ, et al. Landslide displacement prediction based on multi-source data fusion and sensitivity states. *Eng Geology* (2020) 271:105608. doi:10.1016/j.enggeo.2020.105608

62. Wang YK, Tang HM, Wen T, Ma JW. A hybrid intelligent approach for constructing landslide displacement prediction intervals. *Appl Soft Comput* (2019) 81:105506. doi:10.1016/j.asoc.2019.105506

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 24 June 2024
DOI 10.3389/fphy.2024.1419998



OPEN ACCESS

EDITED BY
Zhenqiu Shu,
Kunming University of Science and Technology, China

REVIEWED BY
Lei Yang,
Zhengzhou University, China
Teng Sun,
Kunming University of Science and Technology, China

*CORRESPONDENCE
Weijun Wang,
wj.wang@giat.ac.cn
Zucheng Huang,
zc.huang@giat.ac.cn

RECEIVED 19 April 2024
ACCEPTED 31 May 2024
PUBLISHED 24 June 2024

CITATION
Wang L, Zhang G, Wang W, Chen J, Jiang X, Yuan H and Huang Z (2024), A defect detection method for industrial aluminum sheet surface based on improved YOLOv8 algorithm. *Front. Phys.* 12:1419998. doi: 10.3389/fphy.2024.1419998



# A defect detection method for industrial aluminum sheet surface based on improved YOLOv8 algorithm

Luyang Wang[1,2], Gongxue Zhang[1], Weijun Wang[2]*, Jinyuan Chen[2], Xuyao Jiang[2], Hai Yuan[2] and Zucheng Huang[2]*

[1]College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an, China, [2]Guangzhou Institute of Advanced Technology, Guangzhou, China

In industrial aluminum sheet surface defect detection, false detection, missed detection, and low efficiency are prevalent challenges. Therefore, this paper introduces an improved YOLOv8 algorithm to address these issues. Specifically, the C2f-DSConv module incorporated enhances the network's feature extraction capabilities, and a small target detection layer (160 × 160) improves the recognition of small targets. Besides, the DyHead dynamic detection head augments target representation, and MPDIoU replaces the regression loss function to refine detection accuracy. The improved algorithm is named YOLOv8n-DSDM, with experimental evaluations on an industrial aluminum sheet surface defect dataset demonstrating its effectiveness. YOLOv8n-DSDM achieves an average mean average precision (mAP50%) of 94.7%, demonstrating a 3.5% improvement over the original YOLOv8n. With a single-frame detection time of 2.5 ms and a parameter count of 3.77 M, YOLOv8n-DSDM meets the real-time detection requirements for industrial applications.

KEYWORDS

**defect detection, YOLOv8 algorithm, C2f-DSConv module, DyHead dynamic detection head network, small target detection layer**

## 1 Introduction

The extensive development of technology has led to the widespread use of aluminum and its alloys in diverse industries, including aerospace, transportation, construction, and power generation. However, during the manufacturing process of industrial aluminum sheets, various surface defects, e.g., scratches, pinholes, black spots, and creases, may arise due to the quality of raw materials, production techniques, and equipment conditions. These imperfections compromise the aluminum sheets' aesthetic appeal and, more importantly, diminish their mechanical strength and resistance to corrosion. Consequently, industrial aluminum sheets' usability and service life are adversely affected. Therefore, effectively detecting and controlling surface defects in industrial aluminum sheets is paramount in guaranteeing their quality and reliability.

The surface defect detection of industrial aluminum sheets in production workshops primarily relies on manual visual inspection and tactile methods. However, these approaches are inefficient and heavily influenced by human factors, hindering the detection results' accuracy and consistency. The traditional image detection method consists of three steps: image processing, feature extraction and target recognition. In recent years, with the

advancements in computer vision, image processing, machine learning, and other technologies, deep learning-based surface defect detection methods have gained significant research attention. However, there are still many challenges, such as the slow detection speed of the model, which cannot meet the requirements of real-time detection, the detection environment that affects the detection effect of the model due to factors such as lighting, the low frequency of some defects low, and the small number of available samples. Current deep learning-based object detection algorithms are divided into one-stage and two-stage. The one-stage object detection algorithms, such as the YOLO (You Only Look Once) [1] series and the SSD (Single Shot MultiBox Detector) [2] algorithm, simultaneously locate and classify objects. Accordingly, the two-stage object detection algorithms, such as RCNN [3], Fast R-CNN [4], Faster R-CNN [5], and R-FCN [6], first generate candidate regions and then classify them. Consequently, two-stage object detection algorithms offer higher accuracy but slower detection speed than one-stage algorithms.

For surface defect detection in industrial settings, sacrificing a small portion of detection accuracy and employing a one-stage object detection algorithm is a more practical choice. Currently, several industrial surface defect detection solutions have been proposed. For instance, Sun et al. [7] developed an object detection network based on the R-FCN algorithm for detecting pin-like defects in unmanned aerial vehicle inspection images. They achieved a detection accuracy of 83.45%. Huang et al. [8] proposed an improved aluminum profile surface defect detection algorithm based on Faster R-CNN, which enhanced detection accuracy by incorporating feature pyramids and deformable convolution. However, the detection speed of that method did not meet the industrial requirements, and it consumed a large amount of computing resources. Wei et al. [9] introduced an improved YOLOv3 method for detecting surface defects in steel rolling, utilizing the PSA feature pyramid attention module for multi-scale feature fusion. Their method achieved a detection accuracy of 80.01%. LI et al. [10] developed a lightweight network, M2-BL-YOLOv4, for detecting surface defects in aluminum based on the enhanced YOLOv4. By modifying the backbone network to an inverted residual structure, they significantly reduced the model's parameters and improved its detection speed. Xu et al. [11] proposed an industrial aluminum sheet defect detection method based on YOLOv4, which employed GhostNet [12] as the backbone network to enhance feature extraction capability while reducing network parameters. This approach achieved a detection accuracy of 90.98%. Besides, Tang et al. [13] presented an improved YOLOv5 method for cylinder head forging defect detection, which replaced the SPP-YOLO structure in the original YOLOv5 head with the Decoupled Head structure. This modification enabled the model to utilize multiple feature maps of varying sizes for object detection, adapting to targets with diverse scales. Dou et al. [14] applied an improved YOLOv7 for insulator detection tasks and achieved significant accuracy improvement by incorporating a small target detection layer. Zhou et al. [15] innovatively integrated a context aggregation module (CAM) between the backbone and feature fusion networks based on the YOLOv8 architecture. This approach enhanced feature utilization and yielded a detection accuracy of 89.90% on the photovoltaic cell EL dataset. However, there is still much room for developing surface defect detection technology based on deep learning. For example, defect detection can be fused with cross-modal retrieval technology [16–20], and deep learning models can be used to fuse data from different modalities (images and sensor data) to improve detection accuracy. At the same time, there are still many problems in the surface defect detection of industrial aluminum sheets, such as improving the detection accuracy of the network for small targets, balancing real-time and accuracy, and enhancing the generalization of algorithms.

YOLOv8, a novel algorithm in the YOLO series, introduced by Ultralytics in January 2023, leverages the advancements made throughout the development of the YOLO series to achieve high detection accuracy and speed. Thus, YOLOv8 is well-suited for targeted improvements in surface defect detection of industrial aluminum sheets. Therefore, this paper proposes an industrial aluminum sheet surface defect detection algorithm based on an improved YOLOv8 to enhance defect detection accuracy. Experimental results demonstrate that the improved YOLOv8n algorithm achieves high accuracy on the industrial aluminum sheet dataset while meeting the detection speed requirements for industrial scenarios.

## 2 Introduction to YOLOv8 algorithm

The YOLOv8 algorithm is an advanced object detection model refined and improved upon its predecessors, establishing it as a powerful and highly accurate model. YOLOv8 encompasses five models categorized by size: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The YOLOv8n model was selected as the baseline model due to its compact size and computational efficiency, making it an appealing solution for surface defect detection in industrial aluminum sheets. The structural details of the YOLOv8n algorithm model are illustrated in Figure 1.

The YOLOv8 algorithm comprises three primary components: Backbone, Neck, and Head. The Backbone extracts the features, the Neck performs feature fusion, and the Head is utilized for object classification and localization prediction. YOLOv8 introduces significant innovations and improvements in each component compared to its predecessors. Firstly, YOLOv8 introduces the ELAN concept from YOLOv7 [21] and replaces the previous C3 module with a new C2f module. This modification makes the model more lightweight while obtaining more diverse gradient flow information. Secondly, the Head part adopts a decoupled head structure, using two parallel branches to handle the localization and classification tasks separately, allowing the model to be optimized for different tasks. Thirdly, YOLOv8 replaces the anchor-based approach with an anchor-free one. It also employs the Task-Aligned Assigner sample allocation strategy. Furthermore, YOLOv8 uses the Varifocal Loss for classification and incorporates the Distribution Focal Loss into the original Complete IoU Loss for regression. These modifications enhance the model's generalization capability. Consequently, YOLOv8 emerges as a superior algorithm within the YOLO series, surpassing the performance of most detection algorithms, including YOLOv6 [22] and YOLOR [23].

## 3 Improving the YOLOv8 algorithm

### 3.1 C2f-DSConv module

In order to further enhance the algorithm's detection accuracy, this paper incorporates the DSConv (Dynamic Snake Convolution)

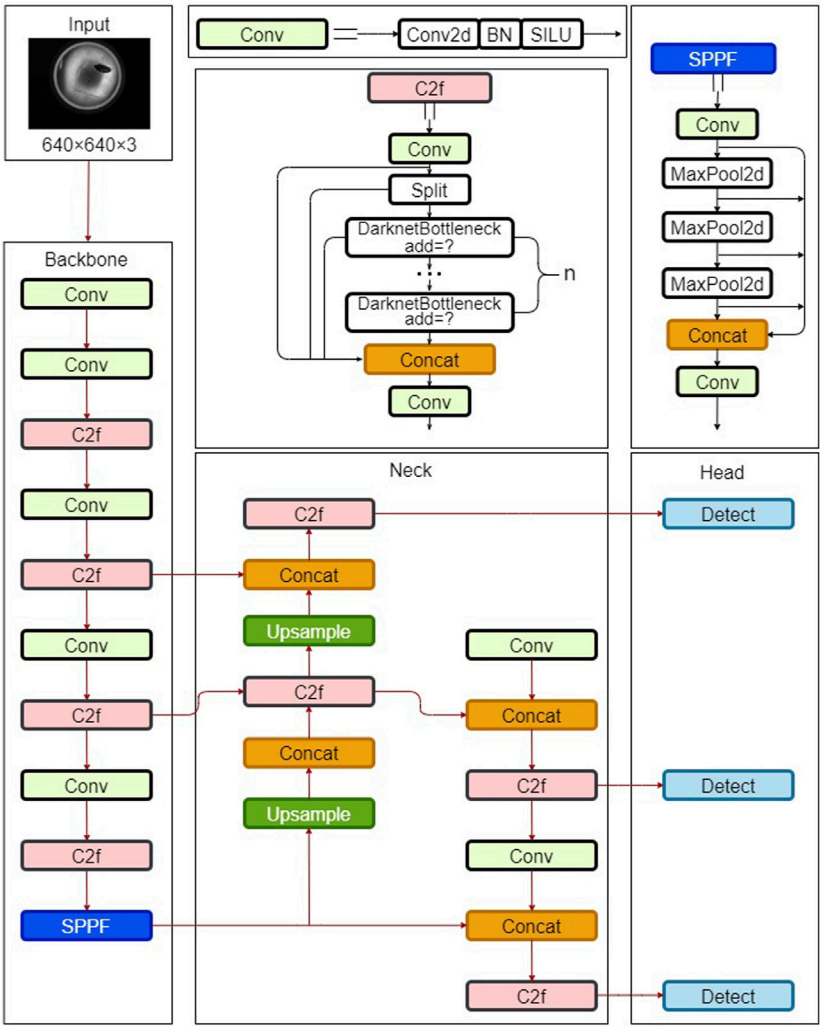



FIGURE 1
Architecture of the YOLOv8n Algorithm model.

[24] module proposed by Southeast University into the C2f module of the YOLOv8 backbone and replaces the two Conv modules in the original C2f module with the DSConv module. Traditional convolutional operations have a fixed receptive field, which can hinder capturing detailed features, particularly locally curved and elongated features, which are challenging to detect due to their limited presence in the image.

Inspired by deformable convolutions, DSConv introduces deformable offsets to traditional convolutions. To prevent the model from learning deformable offsets freely, which could lead to deviations in the receptive field, DSConv employs an iterative strategy. The position of each convolutional operation is determined by using all deformable offsets concerning the central grid as a reference, ensuring the continuity of attention. Figure 2 depicts the DSConv coordinate calculation and the diagram of the receptive field.

Regarding the DSConv coordinate calculation, first, assuming a coordinate K with a size of 3 × 3 for the standard 2D convolution, where the central coordinate is Ki = (xi, yi) and the dilation factor is 1, K can be represented as Eq. (1):

$$K = \{(x-1, y-1), (x-1, y), \ldots, (x+1, y+1)\}, \quad (1)$$

Next, deformable offsets are introduced to enhance the flexibility of the convolutional kernel in capturing the target's complex geometric features. These offsets allow the receptive field to better align with the actual shape of the target. However, to prevent the receptive field from deviating excessively from the target due to unconstrained learning by the model, DSConv applies constraints to the deformable offsets in the *x*-axis and *y*-axis directions. Taking the *x*-axis direction as an example, each grid in K is represented as $Ki \pm c = (xi + c, yi + c)$, $c = \{0, 1, 2, 3, 4\}$. Starting from the center grid Ki, each subsequent grid is incremented by a deformable offset $\triangle = \{\delta | \delta \in [-1, 1]\}$. Since deformable offsets are typically fractional, bilinear interpolation is used. As depicted in Figure 2A, the grid coordinates in the *x*-axis direction are expressed as Eq. (2):

$$K_{i\pm c} = \begin{cases} (x_{i+c}, y_{i+c}) = x_i + c, y_i + \sum_i^{i+c} \triangle y \\ (x_{i-c}, y_{i-c}) = x_i - c, y_i + \sum_{i-c}^{i} \triangle y \end{cases}, \quad (2)$$

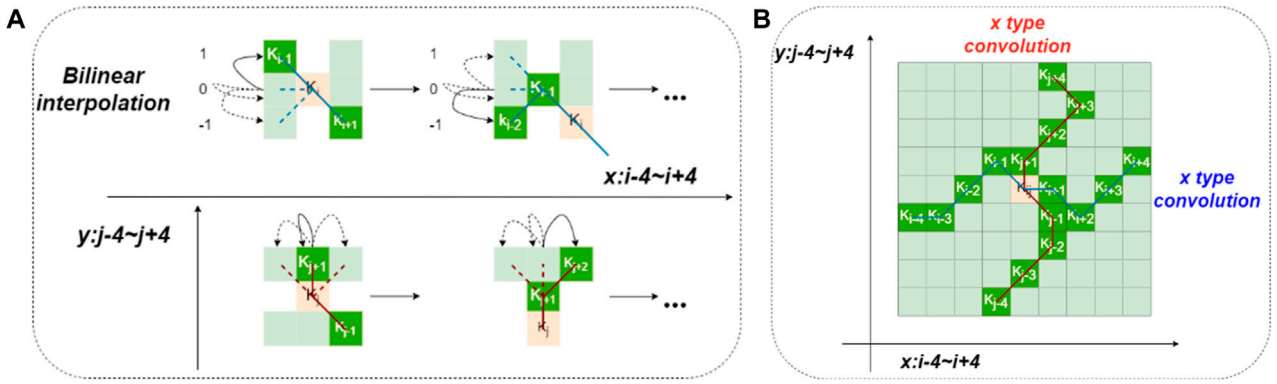



FIGURE 2
**(A)** Coordinate calculation diagram of DSConv, **(B)** DSConv receptive field.

In the *y*-axis direction, the grid coordinates are shown in Eq. (3):

$$K_{j\pm c} = \begin{cases} \left(x_{j+c}, y_{j+c}\right) = x_j + \sum_j^{j+c} \triangle x, y_j + c \\ \left(x_{j-c}, y_{j-c}\right) = x_j + \sum_{j-c}^{j} \triangle x, y_j - c \end{cases}, \quad (3)$$

As illustrated in Figure 2B, the dynamic adjustment of DSConv in the *x* and *y* directions allows its receptive field to cover a range of 9 × 9. Furthermore, DSConv can better adapt to elongated and curved structures from a structural perspective, allowing it to capture more important fine-grained features.

## 3.2 Addition of small object detection layer

Due to uncontrollable factors in industrial environments, aluminum plates often exhibit surface defects, including numerous small objects like holes and tiny scratches. In convolutional neural networks (CNNs), lower-level feature maps have larger dimensions and smaller receptive fields, providing abundant location information and fine-grained features. Conversely, higher-level feature maps have smaller dimensions and larger receptive fields, capturing semantic information [25].

In the original YOLOv8 architecture, the Neck module combines features extracted from the backbone and generates three distinct scales of feature maps to detect objects of varying sizes: 20 × 20, 40 × 40, and 80 × 80. Specifically, the 20 × 20 detects large target objects exceeding 32 × 32, the 40 × 40 medium-sized objects larger than 16 × 16, and the 80 × 80 smaller objects exceeding 8 × 8. When the downsampling factor of the neck is large, the deeper feature map will lose detailed information about the small target, which makes the small target sample difficult to predict.

However, the original YOLOv8 detection layers prove ineffective for detecting minute defects on industrial aluminum plates, often leading to missed detections. Therefore, an additional 160 × 160 detection layer is added at the end of the model to detect tiny objects smaller than 8 × 8. Figure 3 illustrates an upsampling operation added to the neck module after the second upsampling, resulting in a 160 × 160 feature map. This feature map is concatenated with the 160 × 160 feature map from the backbone module, creating a new prediction scale. The modified YOLOv8 model now comprises four detection layers, enhancing its capability to detect small objects.

## 3.3 DyHead

YOLOv8 incorporates the decoupled head structure introduced in YOLOX [26] for classification and localization tasks. However, challenges arise in the industrial aluminum plate inspection process due to variations in defect scales, random changes in angles, and random distributions of defect positions, which the decoupled head struggles to address effectively.

Researchers have investigated how to enhance the detection performance of the Head, with improvements concentrated in three primary areas. 1) Scale perception capability by addressing the presence of targets or defects with vastly different scales within a single image. 2) Spatial perception capability enhances the head's ability to handle targets with varying shapes and positions under different viewpoints. 3) Task perception capability enables the Head to adopt more suitable representation methods for diverse objects.

However, current research focuses on a single aspect of these capabilities. For instance, DyHead (Dynamic Head) [27], proposed by Microsoft, presents a novel dynamic detection Head that simultaneously addresses all three capabilities. Its structure illustrated in Figure 4 reveals that it leverages attention mechanisms in hierarchical, spatial, and channel dimensions, unifying the attention mechanisms of the detection Head, thereby improving detection accuracy and providing a unified analytical perspective for subsequent studies.

DyHead applies the following attention formula (Eq. 4) to the given feature tensor $F \in R^{L\times S\times C}$;

$$W(F) = \pi_C(\pi_S(\pi_L(F)\cdot F)\cdot F)\cdot F, \quad (4)$$

where L, S, and C denote the dimensions of hierarchy, spatial extent, and channel, respectively, $\pi_L$, $\pi_S$, and $\pi_C$ represent the attention functions for these three dimensions $\pi_L$ signifies scale-aware attention, by assigning weights to features of different hierarchical levels based on their semantic relevance for fusion. This is important for detecting objects of different sizes and distances. $\pi_S$ denotes spatial-aware attention, focusing on discriminative regions

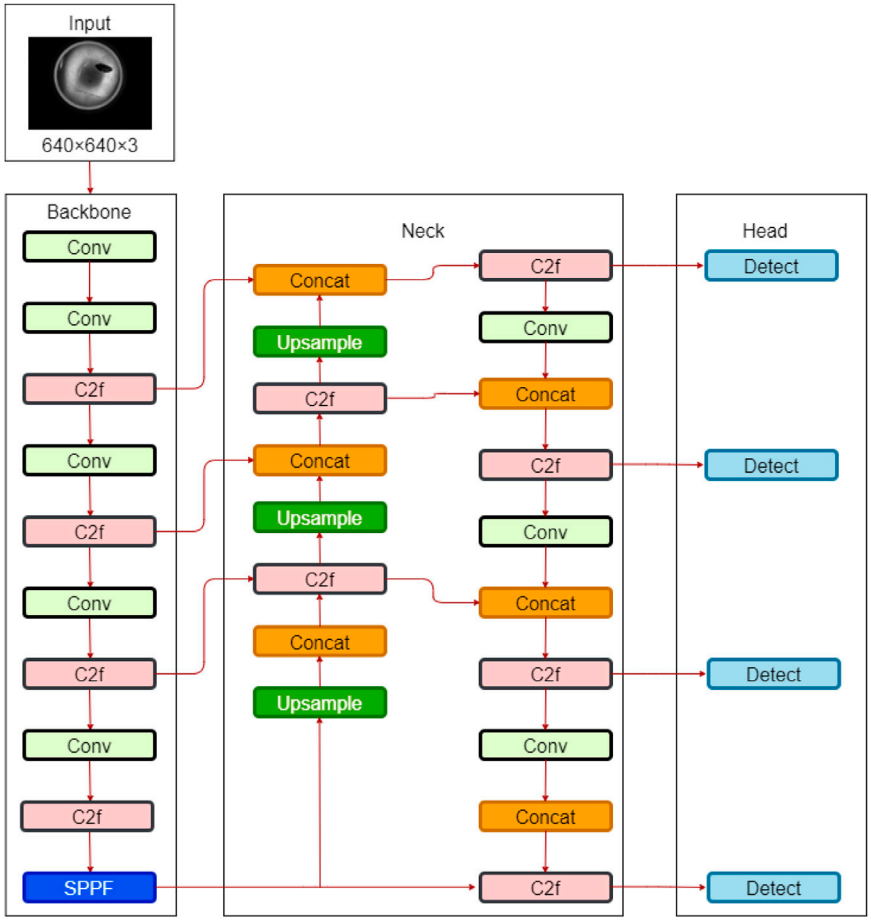



FIGURE 3
Architecture of the YOLOv8 mode with detection layer.

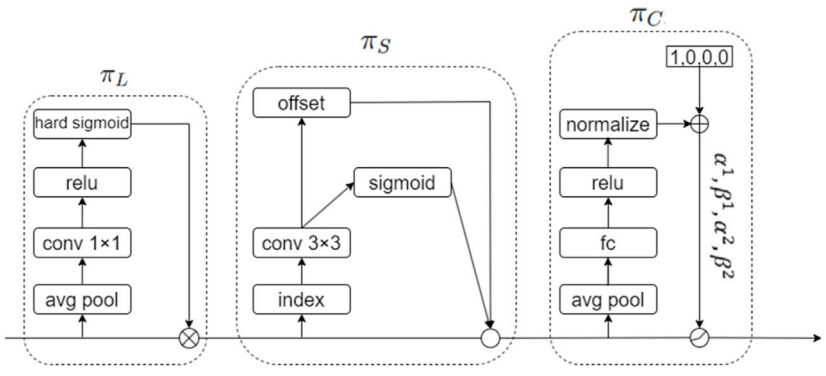



FIGURE 4
Structure of DyHead.

where spatial positions and feature hierarchies align consistently and $\pi_C$ represents task-aware attention, it can dynamically control the ON and OFF of the feature channel to support different tasks, and focus more on the key features of the current task. By unifying different attention perspectives, DyHead significantly enhances the target representation capability of the model with minimal computational overhead.

## 3.4 MPDIoU

YOLOv8's loss comprises two components: classification loss and regression loss. The classification loss evaluates the accuracy of the predicted class, while the regression loss assesses the precision of the predicted bounding box position. Besides, this paper introduces

an improved regression loss function to enhance further the model's detection accuracy. The original YOLOv8 model employs the Complete IoU Loss (CIoU) bounding box loss function [28], which incorporates the impact of aspect ratio based on DIoU. CIoU is formulated as Eqs 5–7:

$$\mathrm{CIoU} = \mathrm{IoU} - \frac{\rho^2(\mathrm{b}, \mathrm{b}^{gt})}{c^2} - \alpha v, \tag{5}$$

$$\alpha = \frac{v}{1 - \mathrm{IoU} + v}, \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \tag{7}$$

where IoU denotes the intersection over union, $\rho^2(\mathrm{b}, \mathrm{b}^{gt})$ is the Euclidean distance between the center points of the predicted and ground truth boxes, and c is the diagonal length of the minimum enclosing rectangle that contains both the ground truth and the predicted boxes. α denotes a weight coefficient, v measures the similarity of aspect ratios, w and h are the width and height of the ground truth box, and $w^{gt}$ and $h^{gt}$ represent the width and height of the predicted box. The CIoU loss is formulated as Eq. (8):

$$\mathrm{LOSS}_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(\mathrm{b}, \mathrm{b}^{gt})}{c^2} + \alpha v, \tag{8}$$

However, the aspect ratio defined in CIoU is a relative value and does not reflect the actual relationship between the width and height of the ground truth and the predicted bounding boxes. This may hinder the model's effective optimization of similarity. Moreover, the bounding box loss function loses effectiveness when the predicted and ground truth bounding boxes have the same aspect ratio but different widths and heights.

Therefore, this paper proposes a new bounding box loss function called MPDIoU (Minimum Point Distance Intersection over Union) [29], which measures the bounding box similarity based on the minimum point distance. Specifically, it directly minimizes the distances between the top-left and top-right points of the predicted and ground truth bounding boxes, thereby simplifying the computation process while considering the non-overlapping area, distance between center points, and width and height deviations. Therefore, MPDIoU can effectively replace CIoU as the bounding box loss function and improve the algorithm's detection accuracy. The MPDIoU and MPDIoU LOSS are formulated as Eqs 9, 10:

$$\mathrm{MPDIoU} = \mathrm{IoU} - \frac{d_1^2}{h^2 + w^2} - \frac{d_2^2}{h^2 + w^2}, \tag{9}$$

$$\mathrm{LOSS}_{\mathrm{MPDIoU}} = 1 - \mathrm{IoU} + \frac{d_1^2}{h^2 + w^2} + \frac{d_2^2}{h^2 + w^2}, \tag{10}$$

where $d_1$ represents the distance between the top-left points of the predicted and ground truth boxes, $d_2$ is the distance between the bottom-right points of the predicted and ground truth boxes, and w and h represent the width and height of the input image, respectively.

The improved network incorporates DSConv into the C2f module of the backbone to enhance the network's feature extraction capability. Additionally, a small object detection layer is added to enhance the network's ability to detect low-resolution small objects, and DyHead is introduced to improve the performance of the detection Head. Finally, the original bounding box loss function is replaced with MPDIoU to improve the algorithm's accuracy. Figure 5 depicts the structure of the improved YOLOv8-DSDM network.

# 4 Experimental setup and results analysis

## 4.1 Dataset

The effectiveness of the YOLOv8n-DSDM algorithm is validated on an industrial aluminum sheet surface defect dataset obtained from the Paddle AI Studio Galaxy Community. All defect images are captured using Hikvision industrial cameras. The dataset comprises 400 images in jpg format, with a resolution of 640 × 480, and involves four types of defects: fold, crake, black, and hole. Each image can contain one or more types of defects, and the total number of defects in the dataset exceeds 1,000. As shown in Figure 6, the dataset sample images illustrate the various types of defects.

This study uses the MVTec Deep Learning Tool annotation software to annotate the four types of defects. The annotated defects are depicted in Figure 7, where yellow, blue, purple, and red represent a fold, crake, black, and hole, respectively.

Furthermore, this study expands the original dataset through data augmentation techniques to overcome the limited sample size of the industrial aluminum sheet dataset and mitigate the risk of overfitting. These techniques include random brightness variation, scaling, Gaussian blur, Gaussian noise, horizontal flipping, random rotation, and vertical flipping. The augmented effects are illustrated in Figure 8. By applying these data augmentation techniques, the total sample size increases from 400 to 3,200, thereby enhancing data diversity and improving the robustness of the deep learning model. To ensure unbiased evaluation, the dataset is divided into training, validation, and testing sets based on a ratio of 7:1:2. Thus, the number of training, validation, and test sets is 2,240, 320, and 640, respectively.

After data augmentation, the specific defect data statistics are presented in Table 1.

## 4.2 Experimental environment and training parameters

The experimental environment and hardware configuration are reported in Table 2, and the training parameters are presented in Table 3. All experiments are conducted under the same experimental environment and parameter settings to ensure validity.

## 4.3 Evaluation metrics

This study uses four evaluation metrics, including mAP@0.5, single-image detection time T, FLOPs, and the number of parameters Params, which are defined as follows:

10.3389/fphy.2024.1419998

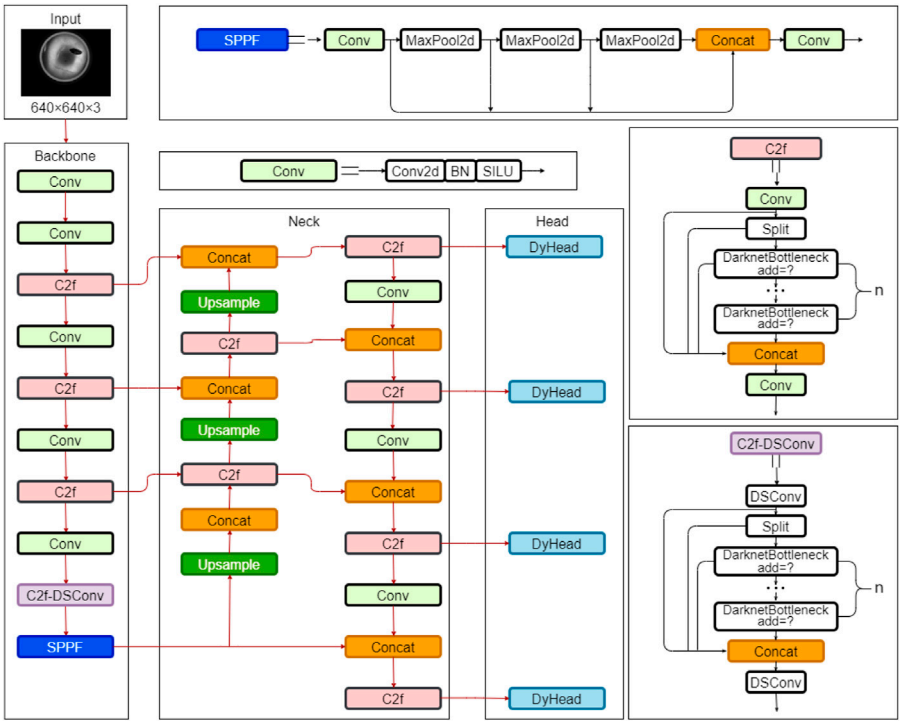



FIGURE 5
Architecture of the improved YOLOv8 algorithm model.

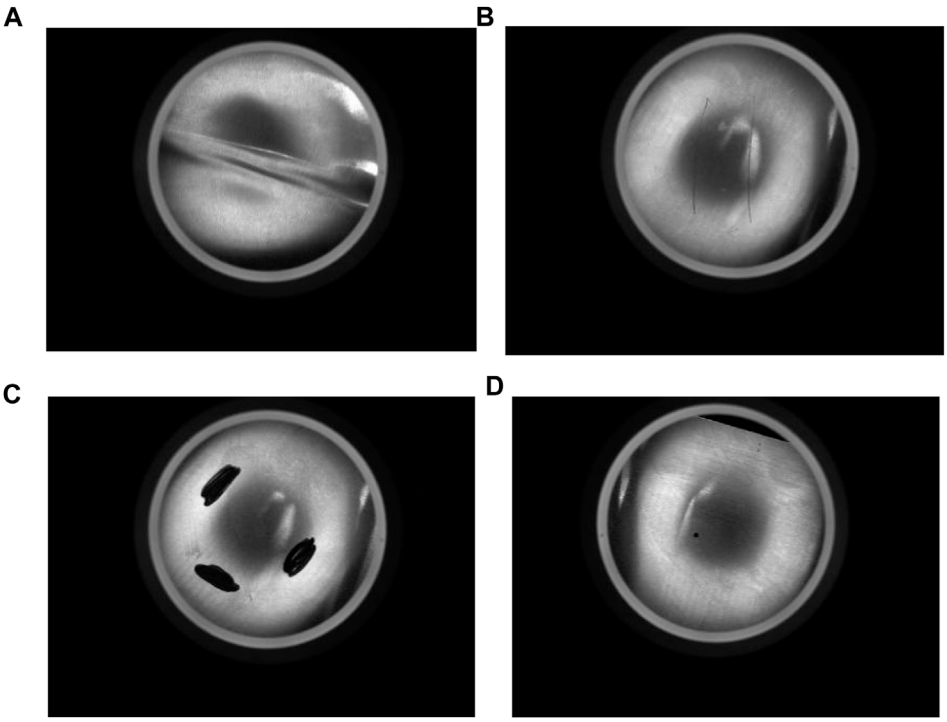



FIGURE 6
Industrial aluminum sheet surface defect Dataset. **(A)** Fold, **(B)** Crake, **(C)** Black, **(D)** Hole.

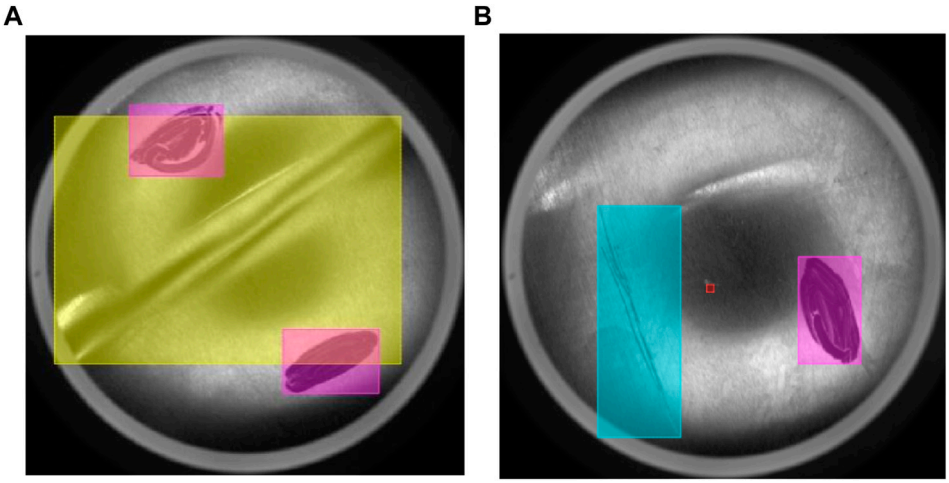



FIGURE 7
Annotated defects. **(A)** Fold and Black, **(B)** Black, Crake and Hole.

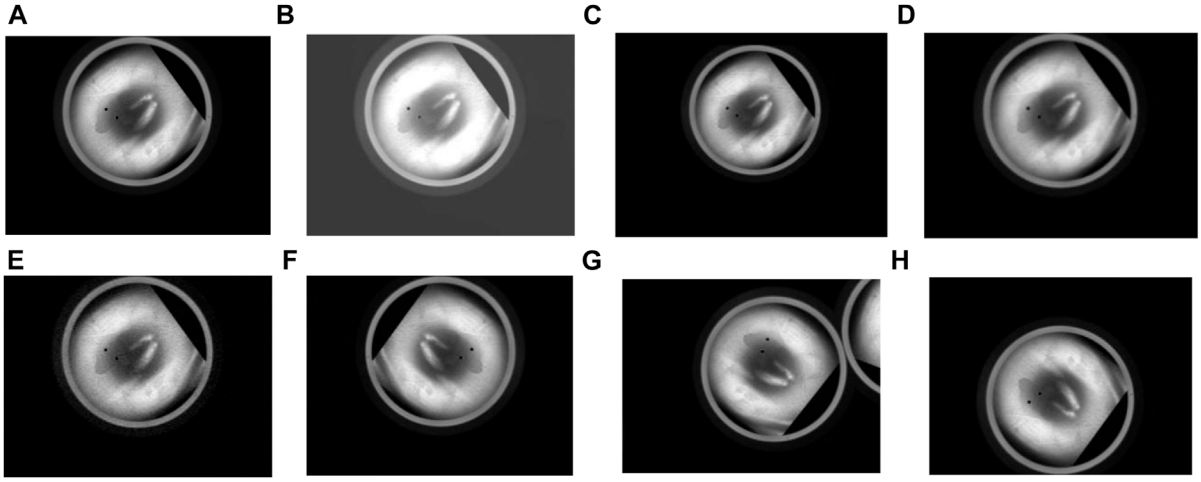



FIGURE 8
Visualization of the augmented data. **(A)** Original **(B)** Random brightness variation **(C)** Scaling **(D)** Gaussian blur **(E)** Gaussian noise **(F)** Horizontal flipping **(G)** Random rotation **(H)** Vertical flipping.

**TABLE 1** Statistical analysis of defective industrial aluminum sheets.

| Defect type | Quantity of each type | | Total | |
|---|---|---|---|---|
| | Before augmentation | After augmentation | Before augmentation | After augmentation |
| Fold | 195 | 1,365 | 1,429 | 10,003 |
| Crake | 475 | 3,325 | | |
| Black | 523 | 3,661 | | |
| Hole | 236 | 1,652 | | |

(1) mAP is the mean average precision, as shown in Eq. (11):

$$\mathrm{mAP} = \frac{1}{\mathrm{class}} \sum_{i=1}^{\mathrm{class}} \mathrm{AP}_i, \tag{11}$$

where class is the total number of categories and $\mathrm{AP}_i$ is the mean average precision of the *i*th category, and mAP@0.5 is the mean of the mean average precision of all categories when IoU is 0.5. The higher the value of mAP@0.5, the better the detection performance of the model.

TABLE 2 Experimental environment parameters.

| Parameter | Value |
|---|---|
| CPU | Intel i9-13900K |
| Memory | 128 GB |
| GPU | NVIDIA RTX4090 *2 |
| Operating System | Ubuntu 20.04 |
| Programming Language | Python3.8 |
| Libraries/Frameworks | PyTorch2.0.1+CUDA11.8 |
| Development Environment | PyCharm |

TABLE 3 Training parameters.

| Parameter | Value |
|---|---|
| Learning Rate | 0.001 |
| Learning Rate Decay Type | cos_lr |
| Total Training Iterations | 500 |
| Batch Size | 32 |
| Optimizer | Adam |
| Optimizer Momentum | 0.937 |
| Weight Decay Coefficient | 0.0005 |

(2) The inference speed of the model is measured based on the single-image detection time T in milliseconds (ms).
(3) FLOPs, or floating-point operations per second, measure computational complexity, reflecting the model's complexity.
(4) Params, or the number of parameters, refers to the total number of trainable parameters in the model, which indicates the model's size.

## 4.4 Ablation experiments of integrating the C2f-DSConv module

Ablation experiments are also conducted on the divided test set to demonstrate the effectiveness of the proposed algorithm and the effectiveness of the C2f-DSConv module in detecting elongated and curved defects on the surface of industrial aluminum plates. In order to ensure the validity of the experiments, the experimental environment and parameter settings were the ones described in the previous section. Table 4 reports the corresponding results.

According to the experimental results presented in Table 4, incorporating the C2f-DSConv module significantly improves the detection accuracy for various defect types. Compared to the baseline model, the accuracy for wrinkles, scratches, dirt, and holes increases by 0.6%, 3.1%, 0.5%, and 1.3%, respectively. Notably, the improvement in scratch detection is the most pronounced, which is important, as scratches on industrial aluminum sheets often exhibit irregular, elongated, and curved shapes, with slender structures occupying a relatively small portion of the image and having limited pixel representation. Moreover, these structures are susceptible to interference from complex backgrounds. Therefore, the experimental results demonstrate that integrating DSConv into the C2f module effectively enhances the model's ability to detect slender and subtle defects. In order to more intuitively demonstrate the effectiveness of adding the C2f-DSConv module to Crake defects, this paper visualizes the feature map in the form of a heat map, which can help us intuitively understand which regions are most important for the model's decision-making. The heat map detection effect is shown in Figure 9. As can be seen from the figure, the model with the C2f-DSConv module pays more attention to the defective part and gives it more weight.

TABLE 4 Ablation experiments of the C2f-DSConv module.

| Model | AP@0.5 | | | | mAP@0.5 |
|---|---|---|---|---|---|
| | Fold | Crake | Black | Hole | |
| YOLOv8n | 94.2% | 84.8% | 97.3% | 88.2% | 91.2% |
| YOLOv8n + DSConv | 94.8% | 87.9% | 97.8% | 89.5% | 92.5% |

## 4.5 Ablation experiments of adding small target detection layer

The following ablation experiments assess the efficacy of the YOLOv8 algorithm enhanced with a small target detection layer. The experimental setup and parameter configurations are the ones previously described. The corresponding findings are reported in Table 5.

In this study, a 160 × 160 small target detection layer is incorporated into the original set of detection layers (20 × 20, 40 × 40, and 80 × 80) in YOLOv8 to identify targets smaller than 8 × 8. The experimental results presented in Table 5 indicate that compared to the original model, the average accuracy for detecting wrinkles increased by 0.4%, scratches by 1.3%, dirt by 0.4%, and holes by 7.4%, with the most notable enhancement observed in hole detection. This outcome can be attributed to the predominance of hole sizes smaller than 8 × 8 on the surface of industrial aluminum sheets, which the original three detection layers struggle to identify effectively. Consequently, these results demonstrate that adding a small target detection layer can significantly enhance the model's detection capability for small targets. At the same time, to more intuitively verify the effectiveness of the hole after adding a small target detection layer, the feature map is displayed in the form of a heat map, as illustrated in Figure 10. As seen in the figure, the model gives more weight to the background before adding the small object detection layer, and after adding the detection layer, the red part of the heat map is mostly concentrated in the defect part to be detected.

## 4.6 Ablation experiments

The proposed model introduces four enhancements to the YOLOv8 model. Hence, six ablation experiments evaluate the effect of these enhancements, including the original model

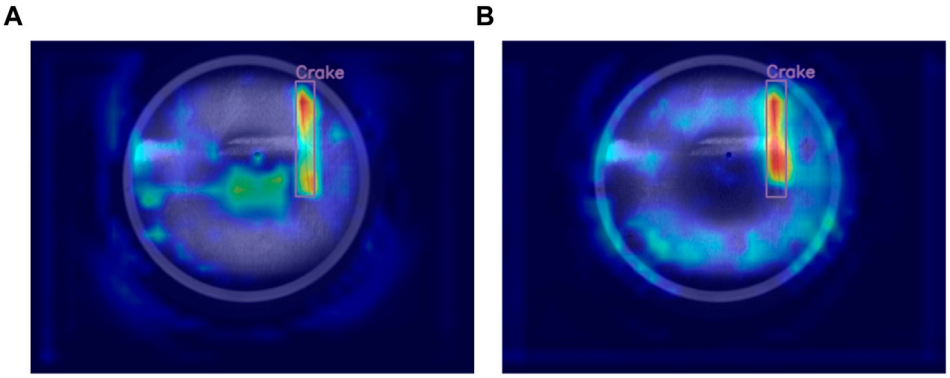



FIGURE 9
Comparison of the heat map effect of the C2F-DSConv module, **(A)** YOLOv8, **(B)** YOLOv8n + C2f-DSConv.

**TABLE 5** Ablation experiments of small target detection layer.

| Model | AP@0.5 | | | | mAP@0.5 |
|---|---|---|---|---|---|
| | Fold | Crake | Black | Hole | |
| YOLOv8n | 94.2% | 84.8% | 97.3% | 88.2% | 91.2% |
| YOLOv8n + Detection Layer | 94.6% | 86.1% | 97.7% | 95.6% | 93.5% |

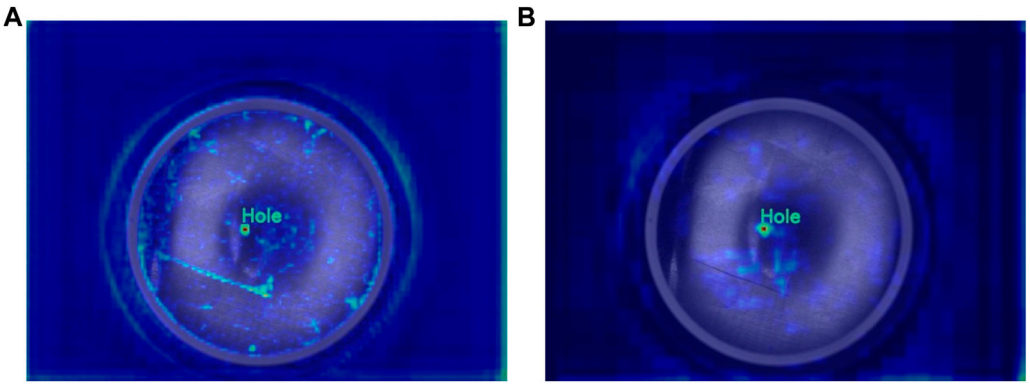



FIGURE 10
Comparison of the heat map effect after adding the detection layer, **(A)** YOLOv8, **(B)** YOLOv8n + detection layer.

experiment, individual implementations of the four enhancements, and their simultaneous integration. The experimental environment and parameter configurations remained constant throughout the trials, and the corresponding findings are summarized in Table 6.

M1 represents the experimental result of the original YOLOv8n model, serving as the benchmark for comparing with other models. It achieves a mAP@0.5 of 91.2%. M2 incorporates the C2f-DSConv module, yielding a 1.3% increase in mAP@0.5 with a marginal rise in the number of parameters, single-frame detection time, and computational load. M3 introduces a small target detection layer, reducing 0.09 M parameters, a 0.3 ms increase in single-frame detection time, and a 4.1G rise in computational load while enhancing mAP@0.5 by 2.3%. M4 integrates the DyHead detection head, leading to a parameter increase of 0.48 M, a single-frame detection time increase of 0.3 ms, and a 1.5G boost in computational load, resulting in a mAP@0.5 increase of 2%. M5 substitutes the boundary box loss function with MPDIoU, maintaining the parameter count and computational load, shortening the single-frame detection time by 0.2 ms, and elevating mAP@0.5 by 0.4%. M6 combines all four improvement methods simultaneously, resulting in a parameter increase of 0.76 M, a 1.2 ms single-frame detection time increase, an 11.6G computational load increase, and the highest mAP@0.5 value of 94.7%. The improved M6 model sacrifices several parameters, single-frame detection time, and computation to provide the highest mAP@0.5 of 94.7%.

TABLE 6 Ablation Experiments of the Proposed Improvement Methods. (√ denotes the use of a specific method and × its non-utilization).

| | C2f-DSConv | Detection layer | DyHead | MPDIoU | Params (M) | T (ms) | FLOPs (G) | mAP@0.5 |
|---|---|---|---|---|---|---|---|---|
| M1 | × | × | × | × | 3.01 | 1.3 | 8.1 | 91.2% |
| M2 | √ | × | × | × | 3.27 | 1.5 | 8.2 | 92.5% |
| M3 | × | √ | × | × | 2.92 | 1.6 | 12.2 | 93.5% |
| M4 | × | × | √ | × | 3.49 | 1.6 | 9.6 | 93.2% |
| M5 | × | × | × | √ | 3.01 | 1.1 | 8.1 | 91.6% |
| M6 | √ | √ | √ | √ | 3.77 | 2.5 | 19.7 | 94.7% |

TABLE 7 Ablation experiments of the improved model.

| Model | AP@0.5 | | | | mAP@0.5 |
|---|---|---|---|---|---|
| | Fold | Crake | Black | Hole | |
| YOLOv8n | 94.2% | 84.8% | 97.3% | 88.2% | 91.2% |
| YOLOv8n-DSDM | 95.7% | 87.7% | 99.2% | 96.2% | 94.7% |

## 4.7 Comparison of detection effects

In order to visually demonstrate the detection performance of the enhanced YOLOv8-DSDM model, a comparative analysis is conducted between the original model and the upgraded model using the test environment and parameter settings reported in Table 7. The corresponding results highlight that the improved YOLOv8-DSDM exhibits enhanced detection accuracy across four types of defects. Specifically, crease, scratch, dirt, and hole detection accuracy increased by 1.5%, 2.9%, 1.9%, and 8%, respectively. Consequently, the average mean accuracy mAP@0.5 increases by 3.5%.

The YOLOv8 and YOLOv8n-DSDM models are challenged on the test set, with the comparative results illustrated in Figure 11. The visual display indicates that the original YOLOv8 model experiences missed detections and imprecise bounding box localization. In contrast, the enhanced model effectively identifies defects overlooked by the original model, leading to closely aligned detection boxes with the targets. Consequently, the YOLOv8n-DSDM model demonstrates an overall superior detection performance.

## 4.8 Comparison with other mainstream object detection algorithms

To assess further the effectiveness of the proposed YOLOv8-DSDM algorithm, comparative experiments are conducted under consistent conditions using an industrial aluminum sheet surface defect dataset. The evaluated algorithms are SSD, Faster R-CNN, DETR [30], RT-DETR [31], YOLOv5 [32], YOLOv6 [22], YOLOv7 [33], YOLOv7-tiny [34], and YOLOv8, with the corresponding results presented in Table 8.

The mAP@0.5 of the SSD algorithm is 67.4%, with a parameter size of 13.69 M. The single image detection time is 1.5 ms, and the computational complexity is 78.20G. The detection accuracy of SSD is relatively subpar, leaving room for optimizing its computational complexity. Comparatively, while the Faster R-CNN algorithm enhances the mAP@0.5 by 5.2%, unlike SSD, it has a considerable increase in parameter size to 27.69 M, accompanied by a surge in single image detection time to 23.9 ms and a substantial rise in computational complexity to 190.83 G. The DETR algorithm, on the other hand, yields a noteworthy 20.8% improvement in mAP@0.5 relative to SSD, with an increase in parameter size of 27.59 M, an increase in single image detection time of 32.8 ms, and an increase in computational complexity of 7.80G. RT-DETR has been optimized based on DETR to achieve real-time object detection, and its mAP@50% can reach 90.4%, and the single detection time is 5 ms. However, the parameters and calculations are still large, which are 28.45 M and 100.6G, respectively. These algorithms have large computational and parameter sizes, making them unsuitable for industrial applications.

Among the YOLO series algorithms, YOLOv5 has the optimal parameter size, single image detection time, and computational complexity, with values of 2.50 M, 1.0 ms, and 7.1G, respectively. YOLOv8 has the highest mAP@0.5, reaching 91.2%. Compared to YOLOv5, YOLOv8 slightly increases mAP@0.5, parameter size, single image detection time, and computational complexity of 1.2%, 0.51 M, 0.3 ms, and 1.0G, respectively. The improved YOLOv8-DSDM algorithm, compared to the original YOLOv8 algorithm, achieves a 3.5% increase in mAP@0.5, reaching 94.7%, at the cost of a parameter size of 0.76 M, single image detection time of 1.2 ms, and computational complexity of 11.6G. In order to see the performance of the proposed model YOLOv8-DSDM more intuitively, the PR curves of each comparison model are given in this paper, as shown in Figure 12. In summary, the proposed YOLOv8-DSDM algorithm outperforms current mainstream algorithms in terms of comprehensive performance.

# 5 Conclusion

This study proposes an enhanced algorithm model, YOLOv8-DSDM, specifically designed to detect defects on industrial aluminum surfaces. This novel model aims to overcome the challenges of low detection accuracy and slow processing speeds associated with conventional methods. Indeed, incorporating DSConv into the C2f module improves the network's feature extraction capacity. Additionally, introducing a 160 × 160 small object detection layer significantly enhances the network's capability to identify small-scale targets. Substituting the original detection

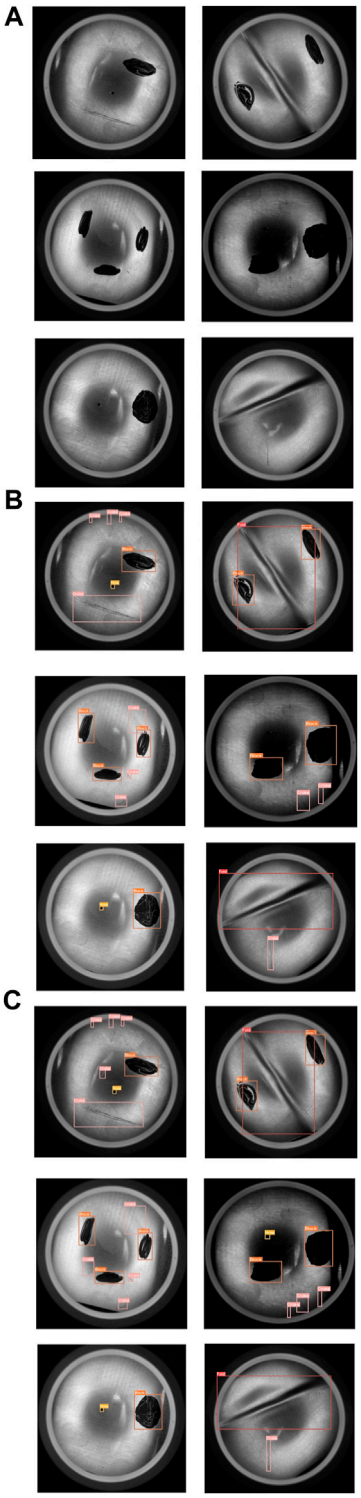



FIGURE 11
Detection performance. **(A)** Test image, **(B)** YOLOv8n, **(C)** YOLOv8-DSDM.

TABLE 8 Comparative experiments with other algorithm models.

| Model | Params (M) | T (ms) | FOLPs (G) | mAP@ 0.5 |
|---|---|---|---|---|
| SSD | 13.69 | 1.5 | 78.20 | 67.4% |
| Faster-RCNN | 41.38 | 25.4 | 269.03 | 72.6% |
| DETR | 41.28 | 34.3 | 86.0 | 88.2% |
| RT-DETR | 28.45 | 5.0 | 100.6 | 90.4% |
| YOLOv5 | 2.50 | 1.0 | 7.1 | 90.0% |
| YOLOv6 | 4.23 | 1.2 | 11.8 | 89.8% |
| YOLOv7 | 36.50 | 4.6 | 103.2 | 87.5% |
| YOLOv7-tiny | 6.02 | 2.1 | 13.0 | 89.5% |
| YOLOv8 | 3.01 | 1.3 | 8.1 | 91.2% |
| YOLOv8-DSDM | 3.77 | 2.5 | 19.7 | 94.7% |

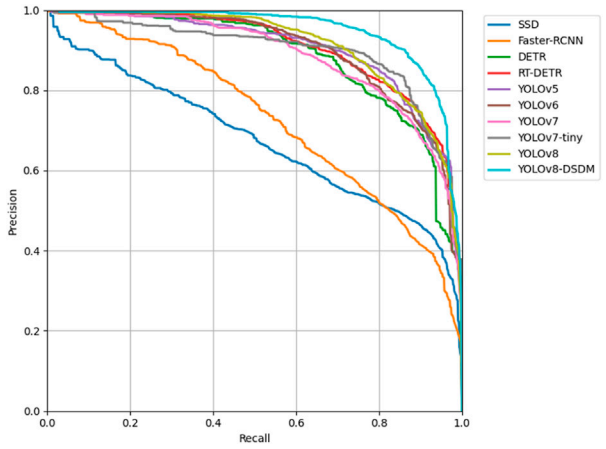



FIGURE 12
PR curve comparison diagram.

head with the dynamic detection head (DyHead) enhances the expressive capacity of the detection head. Moreover, by replacing the original bounding box loss function with the MPDIoU method, we bolster the model's ability to regress bounding boxes while enhancing detection speed. Our experimental findings unequivocally illustrate the substantial advancements in the detection performance of the proposed model. Subsequent efforts will refine the network structure to elevate detection accuracy and streamline model complexity.

## Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

## Author contributions

LW: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Writing–original draft. GZ:

Conceptualization, Data curation, Investigation, Methodology, Project administration, Writing–original draft. WW: Formal Analysis, Funding acquisition, Supervision, Validation, Writing–review and editing. JC: Conceptualization, Data curation, Formal Analysis, Methodology, Resources, Validation, Writing–original draft. XJ: Data curation, Funding acquisition, Project administration, Resources, Supervision, Visualization, Writing–review and editing. HY: Funding acquisition, Investigation, Supervision, Validation, Visualization, Writing–review and editing. ZH: Project administration, Resources, Software, Visualization, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded in part by the National Key Research and Development Program of China (grant number 2018YFA0902900), the Basic Research Program of Guangzhou City of China (grant number 202201011692), and the Guangdong Water Conservancy Science and Technology Innovation Project(grant number 2023-03).

## Acknowledgments

The authors would like to express their thanks to the Guangzhou Institute of Advanced Technology for helping them with the experimental characterization.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note



## References

1. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2017). p. 7263–71.

2. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Computer Vision–ECCV 2016: 14th European Conference; October 11–14, 2016; Amsterdam, The Netherlands. Springer International Publishing (2016). p. 21–37. doi:10.1007/978-3-319-46448-0_2

3. Girshick RB, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR, abs* (2013) 1311:2524. arxiv preprint arxiv:1311.2524.

4. Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision (2015). p. 1440–8.

5. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* (2015) 28. doi:10.1109/tpami.2016.2577031

6. Dai J, Li Y, He K, Sun J. R-fcn: object detection via region-based fully convolutional networks. *Adv Neural Inf Process Syst* (2016) 29. doi:10.5555/3157096.3157139

7. Sun J, Liu G, Liu H. Fine-grained detection of pin defects based on improved R-FCN algorithm and class activation diagram. *Guangdong Electric Power* (2023)) 36(06):50–7.

8. Huang R, Wang J, Zhang D, Sun W. Surface defect detection of aluminum profile based on improved Faster R-CNN. *J Beijing Inf Sci Technology University(Natural Sci Edition)* (2021) 36(05):57–62. doi:10.16508/j.cnki.11-5866/n.2021.05.010

9. Xin W, Yingzi T, Chen L, Zhang G. Improved detection method of rolled steel surface defect of YOLOv3. *Ind Control Comput* (2023) 36(08):85–7.

10. Li S, Guo S, Han Z, Kou C, Huang B, Luan M. Aluminum surface defect detection method based on a lightweight YOLOv4 network. *Scientific Rep* (2023) 13(1):11077. doi:10.1038/s41598-023-38085-x

11. Xu H, Yu H. Industrial aluminum sheet defect detection based on improved YOLO model. *Combined machine tool automatic Process Technol* (2023)(09) 106–111. doi:10.13462/j.cnki.mmtamt.2023.09.023

12. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C, et al. Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2020). p. 1580–1589.

13. Tang Y, Fang K, Yang S, Wang C, Li Z. Research on visual inspection method of cylinder head forging defects improved by YOLOv5. *Manufacturing Technol machine tool* (2023)(08) 166–173. doi:10.19287/j.mtmt.1005-2402.2023.08.024

14. Dou L, Yin X, Ai K, Ma Y. Insulator defect detection algorithm based on improved YOLO. *Electr Technology* (2023)(21) 30–35. doi:10.19768/j.cnki.dgjs.2023.21.008

15. Zhou Y, Yan Y, Chen H, Pei S. Defect detection of photovoltaic cells based on improved YOLOv8. *Adv Laser optoelectronics* (2024) 1–17. Available from: http://kns.cnki.net/kcms/detail/31.1690.tn.20230821.1446.128.html.

16. Yong K, Shu Z, Yu J, Yu Z. Zero-shot discrete hashing with adaptive class correlation for cross-modal retrieval. *Knowledge-Based Syst* (2024) 295:111820. doi:10.1016/j.knosys.2024.111820

17. Li L, Shu Z, Yu Z, Wu XJ. Robust online hashing with label semantic enhancement for cross-modal retrieval. *Pattern Recognition* (2024) 145:109972. doi:10.1016/j.patcog.2023.109972

18. Bai Y, Shu Z, Yu J, Wu XJ. Proxy-based graph convolutional hashing for cross-modal retrieval. *IEEE Trans Big Data* (2023) 1–15. doi:10.1109/tbdata.2023.3338951

19. Shu Z, Bai Y, Zhang D, Yu J, Yu Z, Wu XJ. Specific class center guided deep hashing for cross-modal retrieval. *Inf Sci* (2022) 609:304–318. doi:10.1016/j.ins.2022.07.095

20. Shu Z, Yong K, Yu J, Gao S, Mao C, Yu Z. Discrete asymmetric zero-shot hashing with application to cross-modal retrieval. *Neurocomputing* (2022) 511:366–379. doi:10.1016/j.neucom.2022.09.037

21. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023). p. 7464–7475.

22. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications (2022). arxiv preprint arxiv:2209.02976.

23. Wang CY, Yeh IH, Liao HYM. You only learn one representation: unified network for multiple tasks (2021). arxiv preprint arxiv:2105.04206.

24. Qi Y, He Y, Qi X, Zhang Y, Yang G. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023). p. 6070–6079.

25. Wei C, Yang R, Liu Z, Lan R, Sun X, Luo X, et al. YOLOv8 road scene target detection method with double-layer routing attention. *J Graphics* (2023) 44(6):1104.

26. Ge Z, Liu S, Wang F, Li Z, Sun J. Yolox: exceeding yolo series in 2021 (2021). arxiv preprint arxiv:2107.08430.

27. Dai X, Chen Y, Xiao B, Chen D, Liu M, Yuan L, et al. Dynamic head: unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2021). p. 7373–7382.

28. Zheng Z, Wang P, Ren D, Liu W, Ye R, Hu Q, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans cybernetics* (2021) 52(8):8574–8586. doi:10.1109/tcyb.2021.3095305

29. Siliang M, Yong X. MPDIoU: a loss for efficient and accurate bounding box regression (2023). arXiv preprint arXiv:2307.07662.

30. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European conference on computer vision. Cham: Springer International Publishing (2020). p. 213–229.

31. Liang J, Kong R, Ma R, Zhang J, Bian X. Aluminum surface defect detection algorithm based on improved YOLOv5. *Adv Theor Simulations* (2023) 7:2300695. doi:10.1002/adts.202300695

32. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. Detrs beat yolos on real-time object detection (2023). arxiv preprint arxiv:2304.08069.

33. Dou Z, Hu C, Li Q, Zheng L. Improved surface defect detection algorithm of small sample steel plate in YOLOv7. *Computer Eng Appl* (2023) 59(23):283–292. doi:10.3778/j.issn.1002-8331.2306-0138

34. Haohan L, Yiming F, Huaiqing H, Kanghua H. Improved YOLOv7-tiny's object detection lightweight model. *J Computer Eng Appl* (2023) 59(14). doi:10.3778/j.issn.1002-8331.2302-0115

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 30 August 2024
DOI 10.3389/fphy.2024.1451165



OPEN ACCESS

EDITED BY
Haoyu Chen,
University of Oulu, Finland

REVIEWED BY
Haotian Liu,
University of Oulu, Finland
Imran Iqbal,
New York University, United States

*CORRESPONDENCE
Yuanyuan Wang,
zhfwyy@hyit.edu.cn

RECEIVED 18 June 2024
ACCEPTED 12 August 2024
PUBLISHED 30 August 2024

CITATION
Wang Y, Yan S, Abdullahi HS, Gao S, Zhang H, Chen X and Zhao H (2024) Multiclass small target detection algorithm for surface defects of chemicals special steel.
*Front. Phys.* 12:1451165.
doi: 10.3389/fphy.2024.1451165



# Multiclass small target detection algorithm for surface defects of chemicals special steel

Yuanyuan Wang[1]*, Shaofeng Yan[1], Hauwa Suleiman Abdullahi[1], Shangbing Gao[1], Haiyan Zhang[1], Xiuchuan Chen[1] and Hu Zhao[2]

[1]School of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, Jiangsu, China, [2]Jiangsu Kesheng Xuanyi Technology Co., Ltd., Huai'an, Jiangsu, China

**Introduction:** Chemical special steels are widely used in chemical equipment manufacturing and other fields, and small defects on its surface (such as cracks and punches) are easy to cause serious accidents in harsh environments.

**Methods:** In order to solve this problem, this paper proposes an improved defect detection algorithm for chemical special steel based on YOLOv8. Firstly, in order to effectively capture local and global information, a ParC2Net (Parallel-C2f) structure is proposed for feature extraction, which can accurately capture the subtle features of steel defects. Secondly, the loss function is adjusted to MPD-IOU, and its dynamic non-monotonic focusing characteristics are used to effectively solve the overfitting problem of the bounding box of low-quality targets. In addition, RepGFPN is used to fuse multi-scale features, deepen the interaction between semantics and spatial information, and significantly improve the efficiency of cross-layer information transmission. Finally, the RexSE-Head (ResNeXt-Squeeze-Excitation) design is adopted to enhance the positioning accuracy of small defect targets.

**Results and discussion:** The experimental results show that the mAP@0.5 of the improved model reaches 93.5%, and the number of parameters is only 3.29M, which realizes the high precision and high response performance of the detection of small defects in chemical special steels, and highlights the practical application value of the model. The code is available at https://github.com/improvment/prs-yolo.

KEYWORDS

**object detection algorithms1, steel defects 2, YOLOV83, ParC2Net4, small targets5**

## 1 Introduction

As a key element of the stable operation and safety guarantee of chemical equipment, chemical special steel [1] has excellent corrosion resistance and high temperature and high pressure resistance, and its application value under extreme conditions is incomparable. Whether it's a delicate chemical reactor or a transport line in a harsh environment [2], these steels are essential for efficient and safe industrial production [3]. However, it is precisely this high-intensity application environment that makes even the smallest surface defects, such as small cracks or hidden holes [4], enough to become a potential danger to major safety accidents, directly threatening human safety and environmental protection. Therefore, the development of efficient and accurate defect detection technology has become an urgent problem to be solved, and its urgency and importance are self-evident.

In recent years, the vigorous development of deep learning technology has brought innovation to traditional industries, among which the combination of image recognition technology and deep neural networks is gradually penetrating and reshaping the detection standards of the chemical industry [5]. Given the special complexity of the application scenarios of chemical special steel and the strict requirements for safe production, it is particularly critical to achieve high accuracy in the detection of various minor defects [6]. In practical applications, we have evaluated the existing detection methods (infrared detection method, magnetic flux leakage detection method, etc.), especially in the complex and drastically changing working environment, the missed detection rate of small size defects by traditional means is as high as 30%, and even for small defects less than 1 mm, the missed detection rate rises to nearly 50%. In addition, the traditional method cannot meet the inspection needs of more than 10 workpieces per second in the high-speed production line due to the limitation of reaction speed, which increases the risk of missed detection [7]. This situation cannot meet the needs of high-precision detection, a high recall rate, or accurate positioning for all types of defects in chemical special steel. Therefore, given the limitations and challenges mentioned above, we designed a detection algorithm that can accurately identify a variety of small defects in chemical special steels. This algorithm has high detection accuracy, achieves real-time response and fully meets the comprehensive performance requirements of high identification accuracy, high recall rate, and accurate positioning proposed for defect detection in the field of chemical special steels [8].

In summary, this paper proposes the YOLOv8-based steel defect detection algorithm PRS-YOLO (ParC2Net-RepGFPN-RexSE-Head-YOLO). The contributions of this paper are listed as follows.

- A novel ParC2Net parallel substructure is proposed, which can effectively enhance the capture of local detail features and global information of the target by the backbone network, and improve the detection ability of the model on dense targets on chemical special steels.
- The efficient feature fusion network RepGFPN is adopted, which not only promotes the full interaction between high-level semantic information and low-level spatial information, but also greatly optimizes the transmission efficiency of defect information between various layers and reduces the inference time of the model.
- The MPD-IoU loss function is fused, which optimizes the processing of targets with significant size variation and complex attitude in chemical special steels, effectively enhances the generalization ability of the algorithm, and ensures the high-precision recognition and evaluation performance of the model in complex scenarios.
- A RexSE-Head detection head mechanism is designed, which weights the channel information while improving the parallel processing capability of the detection head, which effectively enhances the sensitivity of the network to small target detection.

# 2 Related

## 2.1 Target detection method

At present, defect identification methods can be summarized into two main categories according to the characteristics of object detection models: one-stage detection and two-stage detection algorithms [9]. The one-stage method, represented by YOLO [10] and SSD [11], has been widely used in industrial defect detection due to its efficient real-time processing speed and practicability. YOLO is particularly suitable for rapid production line monitoring [12] because of its limitations in the identification of small defects and the fact that the positioning accuracy of the bounding box is slightly inferior to that of Faster R-CNN [13]. SSDs improve the detection ability of defects of different sizes by fusing multi-scale feature maps, but their positioning accuracy still faces challenges in the face of extremely small or complex defects, especially in low-contrast backgrounds. On the other hand, the two-stage algorithms, including the R-CNN series and the Mask R-CNN [14], have excellent performance in the accuracy and recall of defect identification due to their step-by-step processing strategies, especially the Faster R-CNN effectively enhances the detection ability of multi-scale defects through RPN [15]. Mask R-CNN introduces instance segmentation on this basis, which greatly improves the depiction accuracy of complex and unknown defect contours, but this improvement in fineness is accompanied by a significant increase in computational cost, which limits its application in scenarios with strict real-time requirements.

In recent years, innovative detection methods have emerged one after another to solve the problem of small target detection, surpassing the traditional two-stage framework, and emerging anchor-free deep models such as PP-YOLOE [16] and Gold YOLO [17], as well as DETR [18] using Transformer architecture. PP-YOLOE optimizes the YOLO design to improve the detection performance of small targets. Gold YOLO's distribution mechanism strengthens the real-time detection accuracy and refreshes the perspective of industrial defect identification. DETR abandons sliding windows and anchor frames to achieve efficient object detection in an end-to-end manner, especially in dense target and long-distance correlation analysis, opening up a new path for small object detection. These cutting-edge technologies not only enrich the inspection methods of chemical specialty steels, but also clarify the future research trend: on the basis of ensuring accuracy, accelerate inspection and save computing resources, and meet the high standards of industrial-grade applications.

In view of the fact that this study focuses on practical application requirements, especially in environments that require fast response and limited hardware resources, the one-stage model is preferred due to its high efficiency. Therefore, the follow-up discussion will deepen the exploration of the optimization path of these models, reveal their potential performance improvement in small object detection through empirical analysis, and integrate the cutting-edge methods mentioned above, such as the anchor-free mechanism based model and high-performance variants, in order to bring more comprehensive and in-depth insights to defect detection technology.

## 2.2 Improved target detection method

In practical application scenarios, in order to achieve efficient, accurate and rapid response detection of small defects (such as cracks, punching, etc.) in chemical special steels, Therefore, to

achieve good results with high detection accuracy and a fast response for small defects such as cracks and punching, it is necessary to carry out targeted optimization of existing defect detection algorithms. Wang et al. [19] adopted the fully convolutional YOLO detection network to conduct an in-depth study of strip surface defects and achieved efficient end-to-end detection. However, with the deepening of the network hierarchy and the application of down sampling operations such as pooling layers, a fully convolutional network may lose some of its spatial position details, resulting in a decrease in the accuracy of fine segmentation of small objects or boundaries. Akhyar et al. [20] optimized the SSD model to identify possible defects on steel surfaces and introduced the RetinaNet method for defect classification. Nevertheless, the SSD model is not ideal for detecting small defects. The default anchor boxes often have difficulty accurately covering and identifying such small targets after multistage down sampling. Xia et al. [21] innovatively improved the YOLO algorithm by adding a coordinate attention mechanism and constructing a feature fusion structure using a multipath spatial pyramid pooling module. Although this improvement enhances the sensitivity of the model to the target position and the detection performance of small targets, it still has the problem of insufficient detection accuracy when facing targets of different scales, complex backgrounds, and sizes.

Kou et al. [22] improved the YOLOv3 algorithm and improved the detection accuracy by introducing a frameless mechanism to improve the detection speed and designing a dense convolutional module to enrich the feature information. Although dense convolutional blocks improve the depth and breadth of feature learning by the model, they also increase the computational complexity and the number of model parameters, which not only consume more storage resources but also may prolong the inference time, especially in the deployment environment of embedded systems with limited resources. In addition, Jiang et al. [23] carefully optimized the YOLOv5 algorithm by using a K-means clustering algorithm to reconfigure the preset anchor parameter to fit the features of actual data samples and added an MA attention mechanism to enhance feature extraction. In addition, the BiFPN module was used to replace the PANet structure in the neck part to achieve comprehensive multiscale feature fusion. These changes improved the detection accuracy by 2.9% while maintaining the lightweight model. However, poor matching between the preset frame and the real target shape can cause defects that cannot be effectively located and identified.

Recent studies, such as the comparative study of automatic image detection and transfer learning [24] and image learning algorithms for small datasets [25], provide valuable references, especially in extracting key features from images and processing small datasets and complex image features.

In view of the existing challenges in the field of defect detection in chemical special steels, especially the limitations of small defect identification, this study innovatively constructs a high-precision multi-category defect detection model, focusing on the accurate detection of subtle defects. By innovating feature extraction, optimizing feature fusion and detection architecture, the model's ability to capture micro-defects and interact with deep features is greatly improved, ensuring excellent positioning and identification performance when dealing with complex defects such as microcracks and fine holes, and fully meeting the high-precision standards for micro-defect detection of special steel in actual production.

# 3 Methodology

While maintaining the advantages of YOLOv5, YOLOv8 is committed to model lightweight and accurate upgrades to adapt to various real-time applications. In this design, the C3 module is abandoned, the C2f module is adopted to strengthen feature extraction and target positioning, and the performance is significantly improved by optimizing the internal integration mechanism [26]. The "head" of the model adopts a decoupled-head design to separate classification and boundary box regression tasks. In the regressive head part, the number of 4*reg_max channels is set by the DFL strategy to enhance the accuracy of position and size prediction and effectively promote the overall prediction efficiency.

Although YOLOv8 has demonstrated powerful real-time detection capabilities in many scenarios, it faces limitations in detecting microscopic defects (such as cracks and punching) in chemical specialty steels. The inherent hierarchical feature extraction mechanism has limited ability to capture small features in low-resolution images, insufficient mining of defect fine morphology and texture information, coupled with the risk of overfitting in cases of strong variability and data scarcity, and the low attention of loss function and optimization strategy to such defects, resulting in limited detection sensitivity and accuracy in this application [27].

To this end, this paper proposes a defect detection model for chemical special steel based on YOLOv8 architecture: PRS-YOLOv8. In response to the complexity of chemical specialty steel defect detection, we adopted histogram equalization, ParC2Net feature extractor, efficient RepGFPN to fuse multi-scale features, and innovative RexSE-Head inspection head design, a series of strategies to ensure that the model can still show excellent real-time inspection accuracy and efficiency in harsh industrial sites. Figure 1 illustrates the comprehensive network architecture design of the PRS-YOLOv8 model.

## 3.1 Data preprocessing

In order to enhance the generalization performance and robustness of the model in complex scenes, we adopted a series of image preprocessing strategies. Firstly, the representation of defects under different viewing angles and sizes is simulated by random scaling, combined with image flipping to reveal the anisotropic characteristics of defects, which effectively alleviates the problem of overfitting and promotes the extensive identification ability of the model. Secondly, the adaptive histogram equalization technology [28] was applied to dynamically optimize the brightness and contrast of the image, especially for the uneven illumination, and effectively suppress the background noise. Unlike the global approach, adaptive equalization processes image areas separately to improve overall image quality while maintaining local contrast, making detailed features more prominent, which is essential for defect detection.

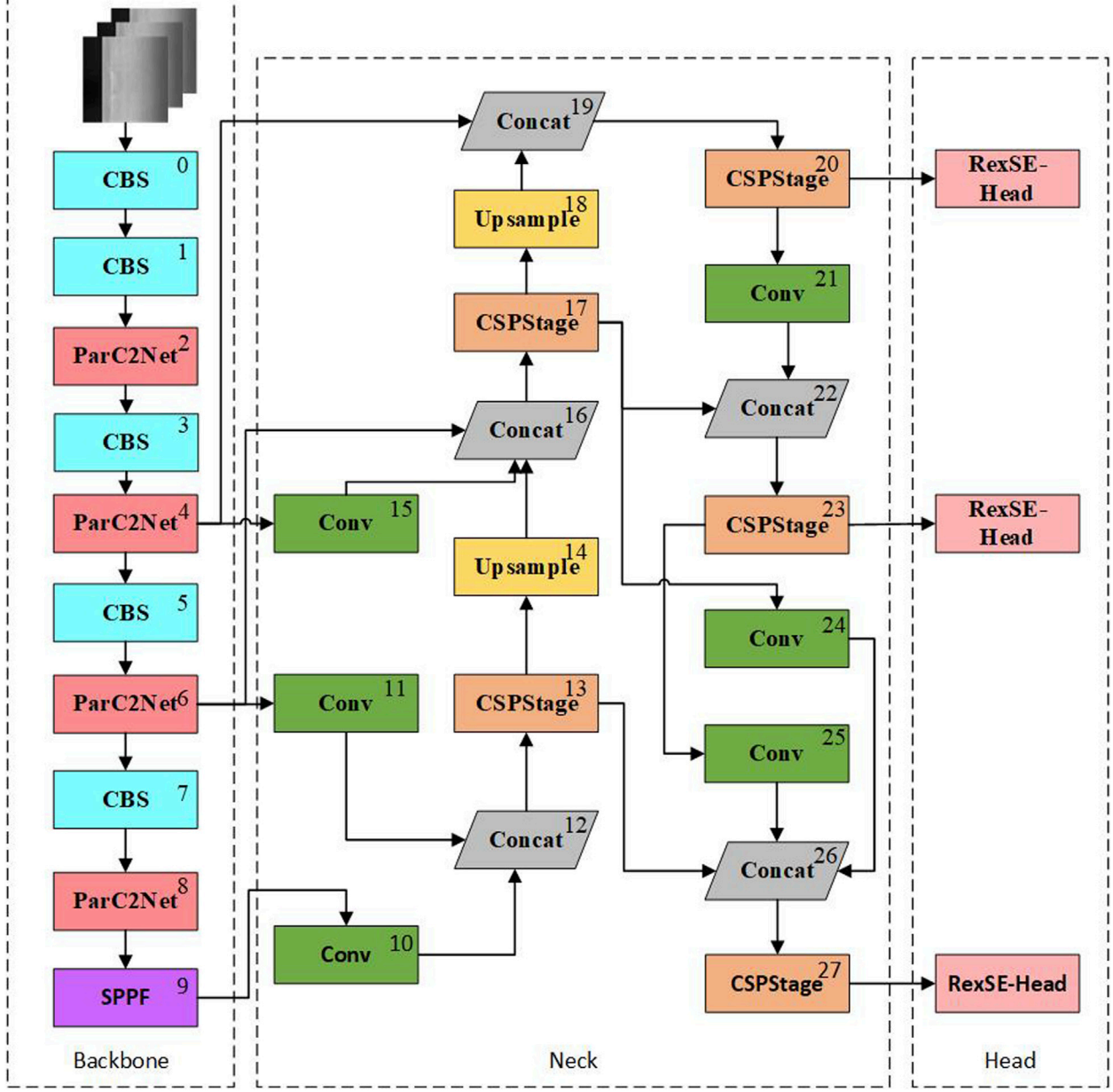



FIGURE 1
PRS-YOLO network structure. The structures of ParC2Net, RexSE-Head, and CSPStage in the neck are explained in detail in 3.2–3.4.

Figure 2 below shows an example of an image after adaptive histogram equalization in the algorithm, which intuitively reflects the role of the technology in enhancing the visual effect of the image and improving the visibility of key details.

## 3.2 Backbone

In the YOLOv8 framework, the backbone component is responsible for extracting key features from the image data, a process that is critical for subsequent defect detection [29]. Although the classical C2f architecture effectively promotes the deep expression of features through the bottleneck building blocks, which integrates the double-layer 3 × 3 convolution and activation function, and enhances the learning potential of the model through residual connection, its understanding of global semantics may inadvertently weaken the focus on subtle local features, which poses an obstacle to the accurate identification of fine defects such as microcracks and punching in chemical special steels, and affects the accuracy of positioning accuracy [30]. In response to this limitation, we innovatively designed the ParC2Net parallel substructure, which is designed to capture multi-scale image details while maintaining the real-time performance of the system.

Specifically, by replacing the bottleneck convolution in the original C2f module with the ParNet architecture, we use its unique parallel flow design to dynamically adjust the size of the

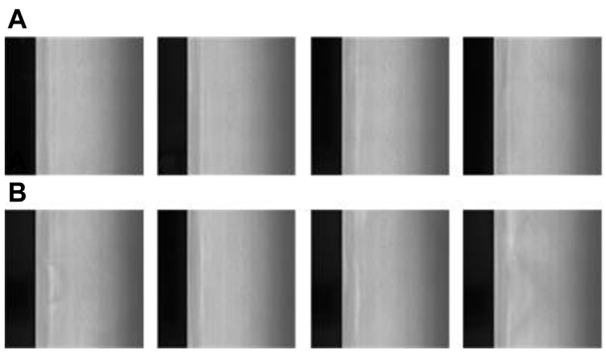



FIGURE 2
Illustration of the effect of adaptive histogram equalization: **(A)** raw image and **(B)** image after adaptive histogram equalization.

receptive field, so as to delicately grasp the local characteristics and global structure information of the defect without sacrificing speed. This dynamic adaptation mechanism of ParNet enables the model to accurately focus on the key regions containing complex details and macrostructure information, which is particularly important for identifying cracks and punching defects with fine local morphology and macrostructure associations. ParNet's core innovation also includes the integrated SSE (Channel Squeeze and Spatial Excitation) attention mechanism [31], which is an advanced feature recalibration strategy. By adaptively learning the weights of different feature channels, the SSE mechanism can enhance the feature expression that is crucial to the detection task, while suppressing irrelevant information, ensuring that the model can clearly distinguish and highlight the decisive features of microscopic defects even in a visually complex background, which greatly improves the feature expression ability and the accuracy of defect detection of the model [32].

As shown in Figure 3, the integration of ParNet not only optimizes the feature extraction process, but also promotes the efficient fusion of feature maps at different levels, realizes cross-scale and multi-dimensional feature capture, and significantly enhances the comprehensiveness and depth of feature extraction. What's more noteworthy is that ParC2Net's simplified architecture design not only ensures high detection accuracy, but also effectively reduces the computing burden and memory occupation, accelerates the inference speed, and ensures that the model can still run efficiently in a resource-limited environment. This feature enables ParC2Net to demonstrate excellent performance stability and adaptability in practical applications dealing with large-scale datasets or hardware resource constraints.

## 3.3 Neck

The neck is the feature pyramid network (FPN), which is responsible for fusing multiscale features from the backbone [33]. By constructing a multiscale feature representation structure, the FPN effectively improves the algorithm's detection performance for objects of different sizes and the model's ability to understand semantic information in complex scenes. However, there are some limitations in the transmission of the one-way information flow of the FPN. To improve the chemical detection ability for dense small target defects in steel, such as cracks and punching, we used RepGFPN [34] to fuse and transmit defect information.

Compared with the traditional FPN structure, the multiscale features of RepGFPN are fused in the two levels of the previous layer and the current layer, which can fully exchange high-level semantic information and low-level spatial information. More importantly, the jump connection of the residual layer provides more efficient information transmission, which can transfer shallow information to deeper structural layers, The architectural details of this process can be clearly seen in Figure 4, which illustrates how RepGFPN optimizes information flow

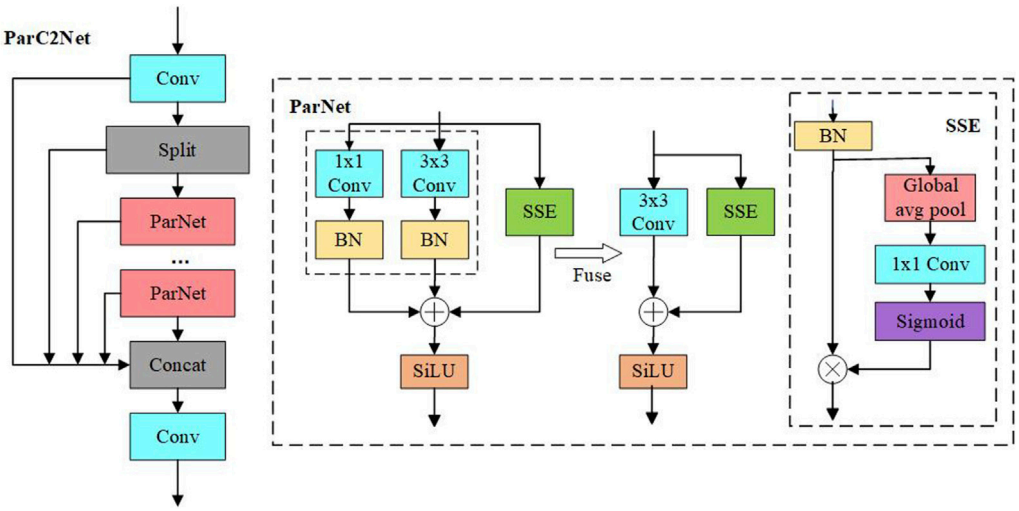



FIGURE 3
ParC2Net network structure. The deep network is built by stacking ParNet Bottleneck modules, and the key features are strengthened with the SSE attention mechanism.

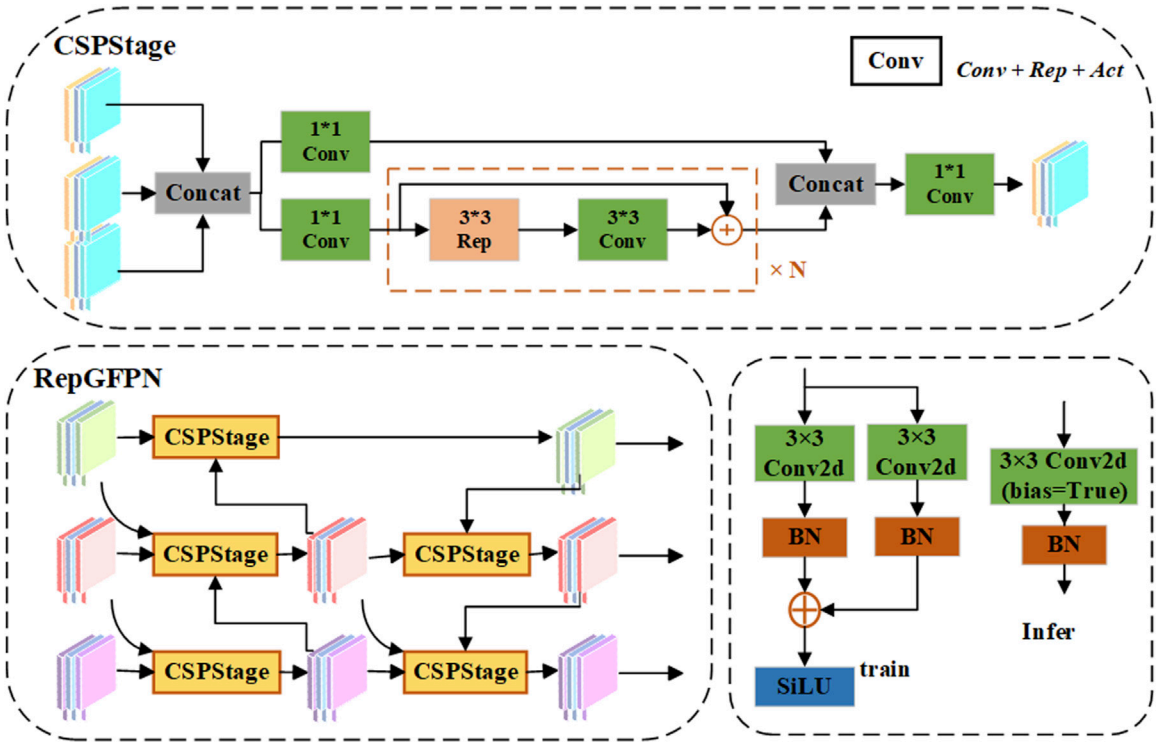



FIGURE 4
RepGFPN structure diagram. The feature information extracted from the backbone network is input to CSPStage, which includes branch, fusion and convolution operations. The Rep module implements the basic RepBlock in the RepVGG and includes training and deployment states.

and feature fusion. In the feature fusion process of the neck, the number of channels in different dimensions corresponding to the feature maps of different sizes is set. By flexibly controlling the number of channels at different scales, higher precision can be achieved by sharing the same channel of all sizes. In the feature fusion module, the CSP stage is used to replace the original feature fusion based on 3x3 convolution. Next, the CSP stage is connected by integrating the heavy parameterization mechanism and the efficient layer aggregation network (ELAN), which achieves higher accuracy without imposing a large additional computational burden. Because small steel targets are usually small in size, subtle in detail, and susceptible to background interference, RepGFPN improves the capture and differentiation of small target features through better feature aggregation capabilities, improving the accuracy of small target detection. Because RepConv uses structural reparameterization, three branches are used for training, and three branches are fused for inference, greatly reducing the inference time. In real-time scenarios, RepGFPN not only achieves efficient frame rates but also improves detection performance, which is particularly important in industrial inspection environments, especially when it is necessary to accurately detect small, fast-moving targets on the production line.

## 3.4 Head

The head is responsible for generating target detection results based on the fused feature map. The head of YOLOv8 consists of multiple output layers, each of which is responsible for detecting objects of different sizes. Due to the low accuracy of defect recognition with small and inconspicuous features, it is necessary to replace the detection head with a more suitable head on a dataset rich in small defects. The RexSE-Head head proposed in this paper improves the detection ability of the model for dense and small targets, especially in scenes where precise capture of microdefects, such as the surface of chemical special steel, is needed.

The core of the RexSE-Head detection head architecture is that the head structure incorporates the ResNeXt [35] and squeeze-and-excitation (SE) attention mechanisms [36]. The specific structure is shown in Figure 5. First, ResNeXt increases the number and width of parallel paths in the network through packet convolution, which improves the parallel processing capability of the detection head and reduces the consumption of computing resources while maintaining high precision. Second, the SE module weights the channel features after each residual block, generates the attention weights of each channel by global average pooling of the feature map, and then learns and adjusts these weights using a two-layer fully connected network. In this way, the model can dynamically adjust the channel contribution degree of the feature graph according to the importance of different parts of the input data, which is conducive to strengthening the attention given to the subtle characteristics of chemical special steel defects and improving the detection performance.

## 3.5 Loss function

The loss function is the core of model training, which quantifies the difference between the predicted output of the model and the

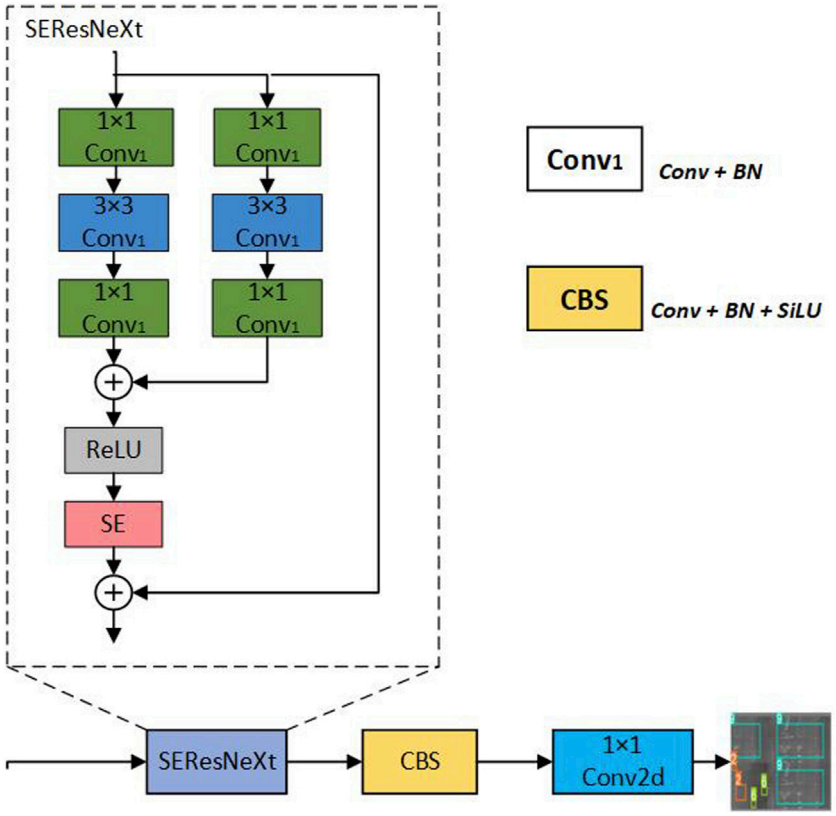



FIGURE 5
RexSE-Head network structure. RexSE-Head is a detection head that is specially designed for chemical special steel defect detection models. RexSE-Head integrates the above ResNeXt bottleneck layer structure with the SE module to improve the network's ability to learn the interactive information between channels to improve the model performance.

actual label, and guides the optimization direction of the model parameters. Specifically, we use a loss function that takes into account a number of key aspects, and its overall framework is defined by Equation 1:

$$L_{all} = \lambda_{\alpha} l_{obj} + \lambda_{\beta} l_{cls} + \lambda_{\delta} l_{box} \tag{1}$$

Among them, $L_{all}$ represents the total loss, which is composed of the confidence loss $l_{obj}$, the categorical loss $l_{cls}$, and the regression loss $l_{box}$ which are constituted by the weighted summation of the balance coefficients $\lambda_{\alpha}$, $\lambda_{\beta}$ 和$\lambda_{\delta}$ to ensure the balanced contribution of each component of the loss.

Confidence loss ($l_{obj}$) Binary cross-entropy is used to measure the degree to which the confidence of each prediction box matches the true existence, and the expression is shown in Equation 2, where $p_i$ represents the true confidence level and $\hat{p}_i$ is the confidence probability predicted by the model.

$$\mathbf{l}_{\text{obj}} = -\sum_{i=1}^{N} \hat{\mathbf{p}}_i \log(\mathbf{p}_i) + (1 - \hat{\mathbf{p}}_i) \log(1 - \mathbf{p}_i) \tag{2}$$

ification loss ($l_{cls}$) also uses the form of binary cross-entropy to evaluate the fit between the predicted class probability distribution and the real class label, as shown in Equation 3, where $y_i$ refers to the actual class label and $\hat{y}_i$ is the class probability distribution predicted by the model.

$$l_{cls} = -\sum_{i=1}^{N} \hat{y}_i \log(y_i) + (1 - \hat{y}_i) \log(1 - y_i) \tag{3}$$

In the regression loss ($l_{box}$) design, we adopted $MPDIOU_{Loss}$ [37] to precisely adjust the position and shape errType equation here.or of the predicted bounding box and the actual labeling box. The mathematical formulation of $MPDIOU_{Loss}$ is detailed in Equations 4–7|. By introducing the concept of MPDIOU (Equation 5), the traditional IoU index is creatively extended to include two distance terms (Equations 6, 7)| based on the normalized bounding box size, i.e., diagonal distance squared $d_1^2$ and $d_2^2$, so as to quantify the deviation between the prediction box and the actual box in terms of spatial layout, and significantly enhance the performance of the model in precise positioning.

$$MPDIOU_{Loss} = 1 - MPDIOU \tag{4}$$

$$MPDIOU = IOU - \frac{d_1^2}{h^2 + w^2} - \frac{d_2^2}{h^2 + w^2} \tag{5}$$

$$d_1^2 = \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2 \tag{6}$$

$$d_2^2 = \left(x_2^{prd} - x_2^{gt}\right)^2 + \left(y_2^{prd} - y_2^{gt}\right)^2 \tag{7}$$

Among them, $(x_1^{prd}, y_1^{prd})$and$(x_2^{prd}, y_2^{prd})$are the diagonal vertex coordinates of the prediction box, while$(x_1^{gt}, y_1^{gt})$and$(x_2^{gt}, y_2^{gt})$ correspond to the corresponding coordinates of the actual box, as shown in Figure 6.

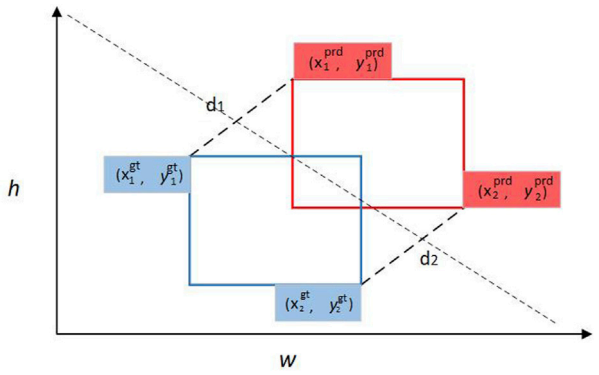



FIGURE 6
Border box crossover ratio -MPDIou.

Compared to the standard IoU, MPDIOU is unique in its mathematical form of non-monotonic focusing, which not only takes into account the measurement of overlapping regions, but also dynamically emphasizes the importance of differences between bounding boxes of different sizes and shapes through the introduction of distance terms. This design allows the loss function to pay more attention to the difficult-to-classify bounding boxes (especially the low-quality target boxes, such as extreme tilt or partial occlusion) during the training process, and effectively alleviates the overfitting problem through a non-uniform loss allocation strategy. Specifically, when the prediction box deviates greatly from the actual box, the $MPDIOU_{Loss}$ will increase significantly due to the increase of the distance term, which forces the model to focus more on the optimization of these difficult cases, and ultimately improves the positioning accuracy and stability of the model on the target boundary in complex scenarios.

# 4 Experimental results and analysis

## 4.1 Experimental dataset

In order to ensure the rigor and reliability of the experimental results, the public dataset GC-DET10 was selected as the benchmark for defect detection of chemical special steels. The dataset contains more than 6,500 images, covering a wide range of defect sizes and balanced number of categories, from tiny defects of less than 1 mm to more obvious damages, while also considering the orientation and orientation of different defects to ensure the diversity of the dataset. The images cover ten common micro-defect types, including Punching (Pu), Weld Line (Wl), Crescent-shaped Gap (Cg), Water Spot (Ws), Oil Stain (Os), Striae (Ss), Inclusions (In), Rolling Pits (Rp), Crease (Cr), and Waist Fold (Wf). It is worth noting that the shape, size and distribution location of defects in the dataset are different, which puts forward high requirements for defect detection algorithms, which need to have excellent generalization ability and robustness to effectively cope with the complex changes of defects under actual working conditions.

In addition, considering the complex lighting conditions that may be encountered in the actual production environment and to further enhance the robustness of the model, we used a variety of data augmentation techniques during the training process. These techniques include, but are not limited to, random rotation, flipping, color dithering, brightness adjustment, and scale shifts to simulate the changes that may be encountered in a real-world production environment. These measures help the model better understand the nature of defect features and maintain high detection accuracy even on unseen samples.

The dataset is scientifically divided into a training set, a validation set, and a test set, with a ratio of 8:1:1, which ensures the rationality of model training, adjustment, and evaluation. Figure 7 visualizes example images of the multiple defect types in the dataset.

## 4.2 Experimental setup

This study relies on a deep learning environment based on a cloud server, with Linux operating system, RTX A6000 GPU, and 51 GB of video memory. The deep learning framework used is Pytorch 2.0, the coding environment runs on Ubuntu 18.04, uses Python 3.10, and uses CUDA version 11.3.

Refer to the official guide of YOLOv8 for the experimental setup, and adopt the free anchor strategy. Table 1 shows the specific parameters.

For the training strategy, we set the initial learning rate to be 0.01, the weight attenuation coefficient to be 0.05, the maximum number of iterations to be 32, and the intersection and union threshold (IoU) to be 0.7. The training process is extended to 200 iterations, and the system automatically performs performance evaluation on the validation set for each epoch learned, so as to continuously monitor the progress of the model and guide the optimization path.

At the same time, in order to ensure the reliability of the training results and effectively reduce the potential bias caused by the randomness of a single experiment, we adopted the following strategies: firstly, the dataset was randomly divided multiple times to generate multiple independent training/validation set combinations; Subsequently, for each division, a complete experimental process and evaluation are rigorously implemented. Finally, the evaluation indicators obtained from each experiment were summarized, and the average value was calculated to obtain a more robust and representative final evaluation result, so as to significantly improve the credibility of data evaluation.

## 4.3 Evaluation metrics

In this article, two key metrics are used to measure model performance: detection accuracy and model size. A number of criteria are used to evaluate detection accuracy, including Recall (R), Precision (P), Average Precision (AP), and mean Average Precision (mAP). Among them, the recall rate reflects the ratio of the identified target to the actual total, The specific mathematical expression is shown in Equation 8:

$$R = \frac{TP}{TP + FN} \times 100\% \tag{8}$$

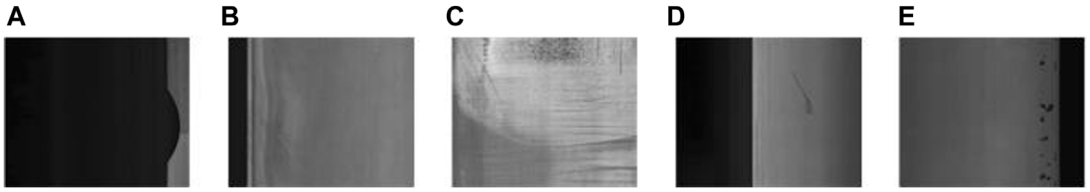



FIGURE 7
Partial types of defects in the dataset: **(A)** half-moon defect, **(B)** inclusions, **(C)** wear defect, **(D)** scratch defect, and **(E)** pitting defect.

TABLE 1 Experimental parameter settings.

| Experimental parameters | Specific values |
|---|---|
| Learning rate | 0.01 |
| Weight decay factor | 0.05 |
| Batch Size | 32 |
| Epoch | 200 |
| IoU threshold | 0.7 |

Here, TP refers to the number of positive samples that are correctly identified (true positives), while FN means the number of positive samples that are not detected (false negatives).

Precision measures the accuracy of a positive sample in a test result and is calculated as shown in Equation 9:

$$\boldsymbol{P} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}} \times \mathbf{100\%} \tag{9}$$

TP is still a true positive, while FP is a negative sample that has been incorrectly classified as a positive sample (false positive).

Average precision (AP) is a comprehensive measure of accuracy at different recall levels, which is obtained by integrating the accuracy within the recall interval, as shown in Equation 10:

$$\boldsymbol{AP} = \int_0^1 \boldsymbol{P}(\boldsymbol{R})\boldsymbol{dR} \tag{10}$$

where P(R) represents the precision of a particular recall level R and dR represents the increment of the recall rate. The process involves determining precision and recall one by one at multiple confidence thresholds, then plotting an accuracy-recall curve and comprehensively evaluating model performance by integrating the region below the curve.

mAP further expands the concept of AP by calculating the arithmetic average of AP values across all classes, ensuring consistency of performance across classes and the validity of the overall evaluation, It is calculated as shown in Equation 11:

$$\boldsymbol{mAP} = \frac{\sum_{t=1}^{N} \boldsymbol{AP_t}}{\boldsymbol{N}} \tag{11}$$

Here, N stands for the number of categories, emphasizing consistency and overall effectiveness of performance across categories.

F-Score is a commonly used performance metric for detection models, which is designed to combine precision and recall metrics. Provide a score that balances the performance of both. It is calculated as shown in Equation 12:

$$\boldsymbol{F - Score} = \mathbf{2} \times \frac{\boldsymbol{Precision} \times \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}} \tag{12}$$

Here, Precision and Recall represent the accuracy of predicting as positive examples and the ability of the model to capture all positive examples, respectively. The F-Score is placed between 0 and 1, and the closer the value is to 1, the better the overall performance of the model.

In addition to evaluating detection accuracy, this paper examines a number of key performance and efficiency metrics such as model size, computational requirements (as measured by Flops), and frame processing speed (Fps). These multiple evaluation dimensions provide valuable insight into the complexity of the model, its computational burden, and its ability to make real-time inferences. In the experimental section of this paper, the methods adopted and the results obtained are described, and the indicators of the model are analyzed and verified.

## 4.4 Test results analysis

### 4.4.1 Model training

During model training, the convergence speed of the loss function slightly represents the performance of the model. We compare the loss function fitting between PRS-YOLOv8 and YOLOv8, and the comparison curves of the two models are shown in Figure 8. With an increase in the number of training iterations, the training curve of the PRS-YOLOv8n model is relatively smooth and can converge to a lower loss level at a faster rate with the same number of iterations. When the loss function value does not change, the training ends, and the loss value of the PRSYOLOv8n model is lower than that of the YOLOv8 model. This finding indicates that the improved model in this paper has better performance than the original model and can more accurately locate and identify target defects.

In order to compare the performance of the model before and after the improvement more clearly, we observed the change trend of Precision, mAP@0.5, Recall, and mAP@0.95 performance indicators with the progress of the training epoch in real time. As shown in Figure 9, the PRS-YOLOv8n has increased accuracy and recall, meaning that it is both accurate and broadly covered when identifying targets, avoiding missed detections. In addition, the significant improvement of the model on mAP, whether it is within the IoU threshold of 0.5 or the range of 0.5–0.95,

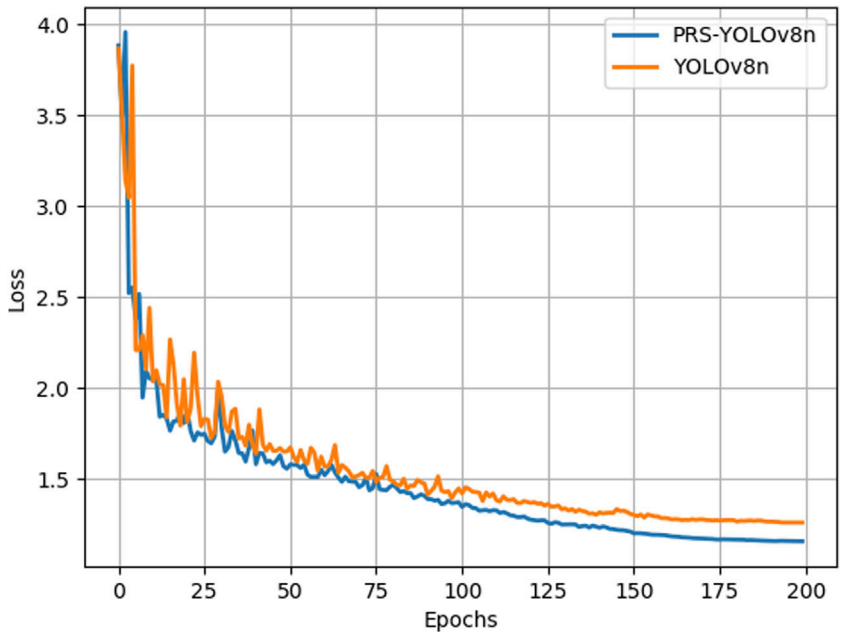



FIGURE 8
Line chart of training loss of PRS-YOLOv8n and YOLOv8n models.

confirms that the model can still maintain excellent detection effect under the diverse matching rigor, highlighting the strong adaptability of the model to scenarios with different accuracy requirements, especially in the early and middle stages of the training cycle, and the superiority of PRS-YOLOv8n is more prominent.

In general, the improved YOLOv8n detection model in this paper has high accuracy and good detection performance, which can better meet the application requirements of chemical special steel defect detection.

### 4.4.2 Detection effect of different defects

To verify the ability of the model to detect ten common minor defects in chemical special steel, the performances of the YOLOv8 model and PRSYOLOv8 model were evaluated in terms of the mAP@0.5, precision, mAP@0.95, and recall. A comparison of the performances of the two models is shown in Figure 10.

Experiments show that compared with the YOLOv8 model, PRS-YOLOv8 can significantly improve the precision and recall of punching, crescent-shaped gap, water spot, waist fold, and other small target defects. This finding indicates that the PRS-YOLOv8 model has greater localization and recognition ability for small target defects in chemical special steel. When dealing with defect categories with high texture similarity, such as Oil Stain and Striae, the model shows a significant improvement in detection accuracy, which strongly proves that it has stronger resolution and accuracy in the recognition and classification of defects of the same nature. This improvement not only improves the accuracy of the detection algorithm, but also demonstrates the excellent performance of the model in complex texture recognition and fine classification.

### 4.4.3 Ablation experiment

To verify the validity of each component of the proposed method, corresponding ablation experiments are performed on each branch in this paper. The experimental results are shown in Table 1, among which ParC2Net represents the designed parallel architecture, RepGFPN represents the feature pyramid network used by the neck, and RexSE-Head represents the designed detection head mechanism. The baseline network model that was adopted is the YOLOv8n network.

Table 2 shows that the proposed method significantly improves the detection performance when it gradually introduces the ParC2Net, RepGFPN, and RexSE-Head structures. Compared with that of YOLOv8n, the precision of ParC2Net increases by 3.7%, indicating that the parallel flow design of this structure can improve the backbone network's ability to extract minor defect information by 1.2% mAP@0.5% and 2.1% mAP@0.95. This finding indicates that the average precision of the model increased under different IoU thresholds, especially the high threshold, confirming that ParC2Net can improve the model's ability to identify and locate small targets by increasing attention to important features. Second, when the RepGFPN module is introduced, the recall rate is increased by 3.1%, and the precision is increased by 1.9%, which indicate that the deep fusion of semantic information can effectively reduce the probability of missing small and medium defects of chemical special steel and has a positive effect on improving the identification accuracy of the detection model. The application of the RexSE-Head detection head achieved a performance improvement with an accuracy of 1.2% and a recall rate of 2.3%, which highlighted the ability of the algorithm to efficiently capture targets of different scales, especially small defects, and confirmed that by widening the parallel path and adjusting the weight of the output feature channel, the algorithm can effectively improve the accuracy of locating and

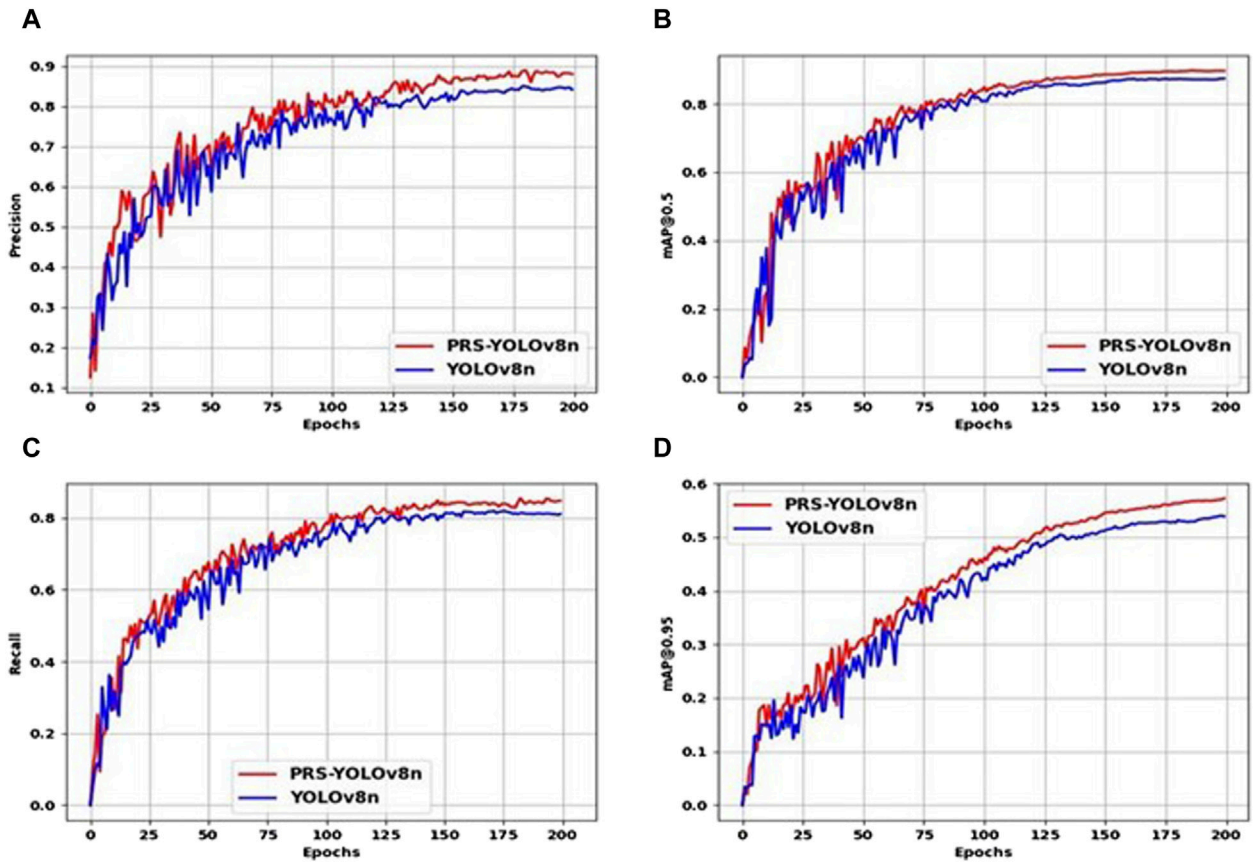



FIGURE 9
Comparison chart of the real-time performance of YOLOv8n and PRS-YOLOv8 training. **(A)** Precision comparison chart. **(B)** mAP@0.5 Comparison chart. **(C)** Recall comparison chart. **(D)** Comparative chart mAP@0.95.

classifying defects. Combining the two indices mAP@0.5 and mAP@0.95, the three modules increase mAP@0.5 by 1.2%, 1.7%, and 0.9% and mAP@0.95 by 2.1%, 2.2%, and 1.1%, which proves the effectiveness of these components once again.

To further understand how these components affect model performance, we used Grad-CAM technology to visualize the key areas of focus of the model. Figure 11 shows the Grad-CAM heat map, where the red areas indicate the parts of the model that are of focus when performing small defect detection. From these heat maps, we can observe how ParC2Net, RepGFPN, and RexSE-Head work together to guide the model to focus on those feature regions that are critical for small object detection.

From these heat maps, it can be seen that the ParC2Net structure can effectively capture the subtle features around the defect, the RepGFPN module helps the model understand the global context of the defect, and the RexSE-Head strengthens the model's ability to identify the key features of the defect. These heat maps provide visual evidence of the important role these three components play in improving small defect detection performance.

### 4.4.4 Comparative test

To evaluate the defect detection performance, the PRS-YOLOv8 algorithm is compared with five target detection algorithms: SSD, YOLOv5, YOLOXs, DETR, and Faster R-CNN. To verify the superiority of the model from multiple angles, the experiment adopts three model sizes of n, s, and m for comparison. Standard evaluation indices such as parameter number, average accuracy (mAP), recall, and FPS were selected to comprehensively evaluate the performance of different algorithms in chemical steel defect detection. The hardware facilities and datasets used were consistent. The final experimental results are shown in Table 3.

The experimental data show that compared with common target detection algorithms, the PRS-YOLOv8 model has distinct advantages in defect detection for chemical special steel. First, compared with that of YOLOv8n, the parameter number of PRS-YOLOv8n increased by only approximately 0.14 M, but the index of mAP@0.5 increased by 2.1%. Notably, when the IoU threshold is 0.95, the mAP increases by 3.4%. This finding indicates that the improved model not only achieves higher detection accuracy with a small number of parameters but also greatly improves the detection and positioning accuracy of small objects.

Secondly, as shown in Figure 12, in the horizontal comparison of various size models, the PRS-YOLOv8 series designed by us surpasses the basic YOLOv8 model in the n, s, m, and l versions, demonstrating better mAP performance. Although PRS-YOLOv8 has made some concessions in terms of operating speed (FPS), it still has a significant advantage in competition with traditional algorithms such as Faster R-CNN, and has achieved significant growth in the high-precision standards, namely, mAP@0.5 and mAP@0.95, which highlights the deep optimization of detection

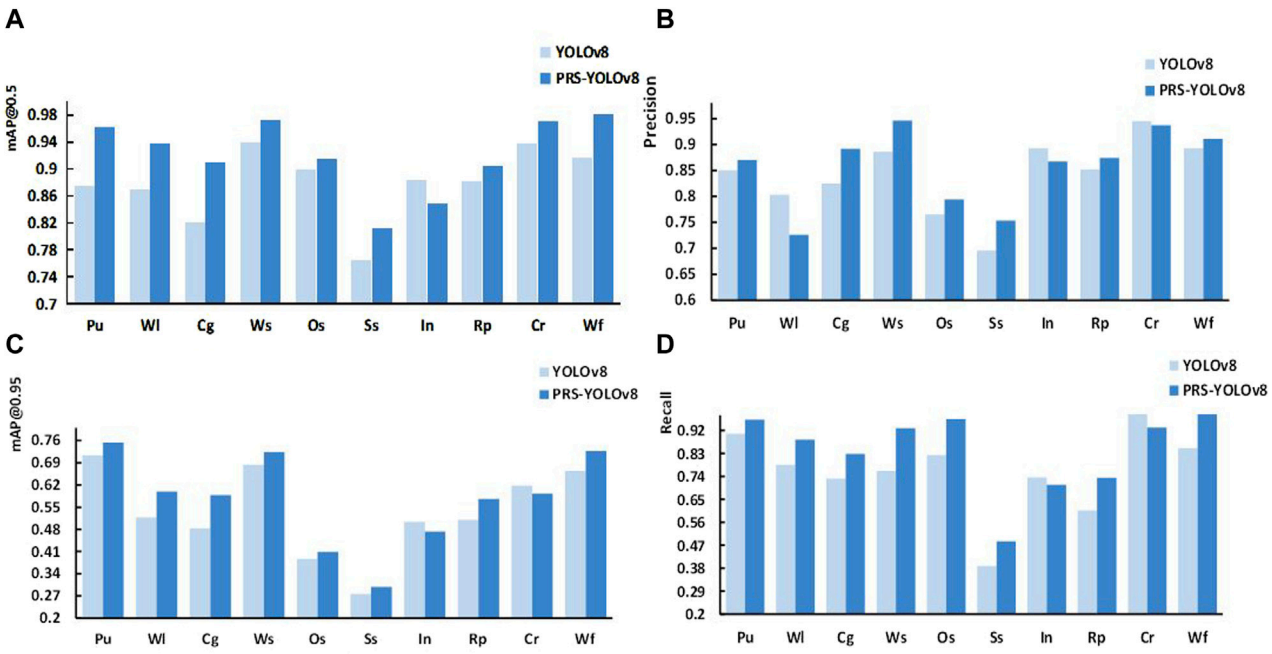



FIGURE 10
Comparison of the detection results for different types of defects between YOLOv8 and PRS-YOLOv8: **(A)** comparison chart of mAP@0.5, **(B)** precision comparison chart, **(C)** mAP@0.95 precision chart, and **(D)** recall comparison chart.

**TABLE 2** Results of the ablation experiment.

| Model | Precision | Recall | mAP@0.5 | mAP@0.95 |
|---|---|---|---|---|
| YOLOv8n | 84.1% | 81.1% | 87.5% | 53.8% |
| YOLOv8n + ParC2Net | **87.8% (+3.7)** | 81.7% | 88.7% | 55.9% |
| YOLOv8n + RepGFPN | 86.0% | 84.2% | 89.2% | 56.0% |
| YOLOv8n + RexSE-Head | 85.3% | 83.4% | 88.4% | 54.9% |
| PRS-YOLOv8n | 87.9% | **84.7% (+3.6)** | **89.6% (+2.1)** | **57.2% (+3.4)** |

The best experimental results are marked in bold, and the values in parentheses reflect the gain of the comparison base model.

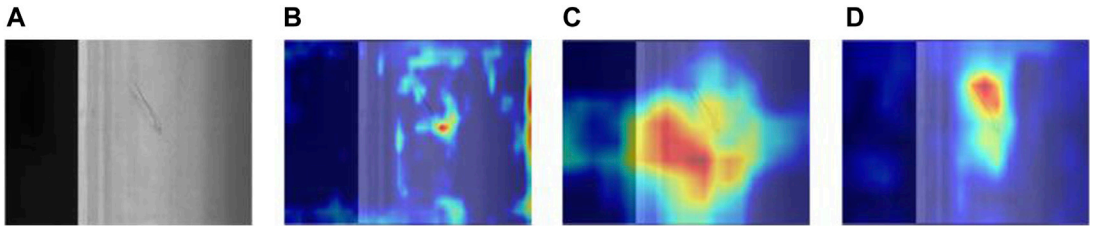



FIGURE 11
The area of focus for each component when detecting small defects. **(A)** Original image of steel Striae. **(B)** Feature heat map using ParC2Net. **(C)** Characteristic heat map using RepGFPN. **(D)** Feature heat map using RexSE-Head.

accuracy by PRS-YOLOv8 while maintaining efficient inference rates. In order to further verify the advantages of the PRS-YOLOv8 model over other advanced object detection algorithms, we compare it with recent algorithms designed for small object detection, including Gold YOLO [17], EfficientDet-D0 [40], and the latest DAMO-YOLO-L [34] and PP-YOLOE-L [16] models. A relatively low computational complexity (measured in GFLOPs) is maintained. This means that PRS-YOLOv8 can achieve better

TABLE 3 Comparative experimental results.

| Model | Backbone | Params (M) | GFLOPs | FPS | mAP@0.5 | mAP@0.95 |
|---|---|---|---|---|---|---|
| YOLOv5-n [38] | — | 2.5 | 7.2 | 121.6 | 84.8% | 51.1% |
| YOLOv5-s | — | 9.15 | 24.2 | 122.0 | 90.6% | 64.4% |
| YOLOv5-m | — | 25.1 | 64.6 | 102.8 | 93.7% | 72.4% |
| YOLOv5-l | — | 46.5 | 119.6 | 96.5 | 94.2% | 73.6% |
| YOLOv8-n | — | 3.15 | 8.7 | 111.2 | 87.5% | 53.8% |
| YOLOv8-s | — | 11.16 | 28.6 | 133.1 | 91.9% | 65.2% |
| YOLOv8-m | — | 25.9 | 78.9 | 107.2 | 95.6% | 74.4% |
| YOLOv8-l | — | 43.7 | 165.2 | 92.3 | 96.2% | 75.6% |
| YOLOX-s [39] | — | 9.0 | 26.8 | 137.5 | 78.9% | 44.7% |
| YOLOX-m | — | 25.3 | 73.8 | 150.3 | 90.5% | 56.8% |
| YOLOX-l | — | 54.2 | 155.6 | 112.0 | 92.1% | 62.0% |
| YOLOX-x | — | 99.1 | 281.9 | 98.2 | 93.8% | 67.5% |
| Gold YOLO-n [17] | — | 5.6 | 12.1 | — | 82.5% | 54.9% |
| Gold YOLO-s | — | 21.5 | 46.0 | — | 90.1% | 57.5% |
| Gold YOLO-m | — | 41.3 | 87.5 | — | 93.5% | 63.4% |
| Gold YOLO-l | — | 75.1 | 151.7 | — | 95.7% | 70.5% |
| **PRS-YOLOv8-n** | — | 3.29 | 9.2 | 57.4 | **89.6% (+2.1)** | **57.2% (+3.4)** |
| **PRS-YOLOv8-s** | — | 12.2 | 29.7 | 77.9 | **94.9% (+3.0)** | **67.7% (+2.5)** |
| **PRS-YOLOv8-m** | — | 29.5 | 82.6 | 78.9 | **96.9% (+1.3)** | **75.0% (+0.6)** |
| **PRS-YOLOv8-l** | — | 37.4 | 143.5 | 65.2 | **97.3% (+1.1)** | **78.2% (+2.6)** |
| SSD [11] | — | 2.4 | 59.6 | 98.7 | 76.4% | 39.1% |
| Faster RCNN [13] | R50-FPN | 42.0 | 930.7 | 18.5 | 85.2% | 48.7% |
| DETR [18] | R50 | 41.0 | 187 | - | 83.2% | 43.3% |
| DAMO-YOLO-L [34] | — | 42.1 | 97.3 | 126 | 87.5% | 65.5% |
| PP-YOLOE-L [16] | — | 52 | 110 | 94 | 88.9% | 67.6% |
| EfficientDet-D0 [40] | — | 3.9 | 7.8 | — | 83.0% | 52.1% |

The results of the multi-size (n, s, m, l,) experiments of the design model in this paper are highlighted in bold, and the values in parentheses show the performance improvement compared to the base model.

detection results with lower resource overhead in actual deployment scenarios, which undoubtedly lays a solid foundation for its application in resource-constrained environments, and fully reflects the excellent design of the model in terms of balance between efficiency and accuracy.

In summary, the PRS-YOLO model has shown strong competitiveness and wide application potential in defect detection of chemical special steel products from the perspective of detection accuracy, computational complexity, and real-time performance.

### 4.4.5 Visual result analysis

In this study, a heat map was utilized to visualize the results of defect detection. By observing the highlighted areas in the heat map, you can visually assess the detection capability of the model and the accuracy of target positioning. The experimental results are shown in Figure 13.

According to a comparison of the defect heatmaps of the YOLOv8 model and the PRSYOLOv8 model in Figure 13, the PRS-YOLOv8 model shows more obvious attention to the defect target. In addition, the comparison results demonstrate the accurate location and identification of the defect object. This finding shows that the PRS-YOLOv8 model effectively captures the key features of the special steel defect detection task, thus achieving accurate boundary box prediction.

### 4.4.6 Test results

The purpose of this experiment is to comprehensively evaluate the performance of the PRS-YOLOv8 model on the test dataset, with special attention to its ability to identify different defect categories,

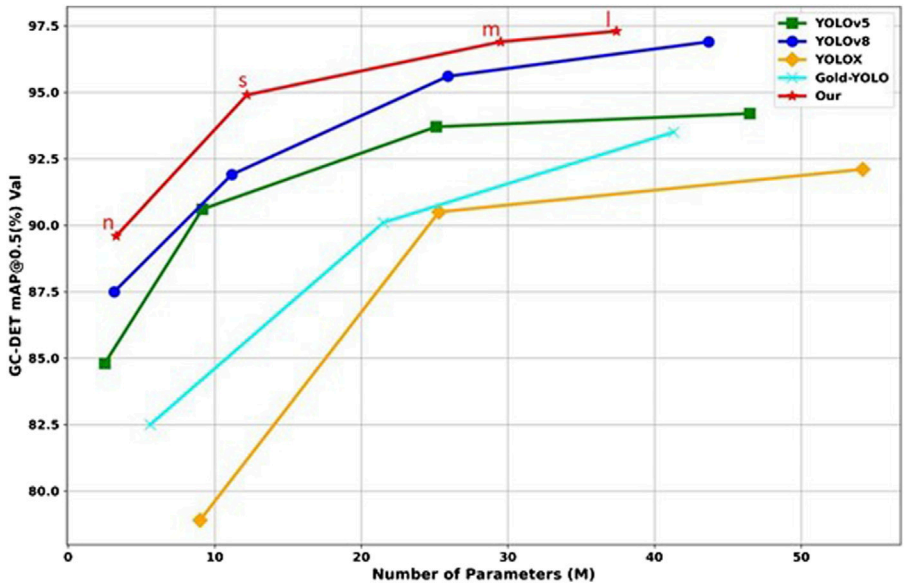



FIGURE 12
This article compares the model with the most advanced real-time object detectors.

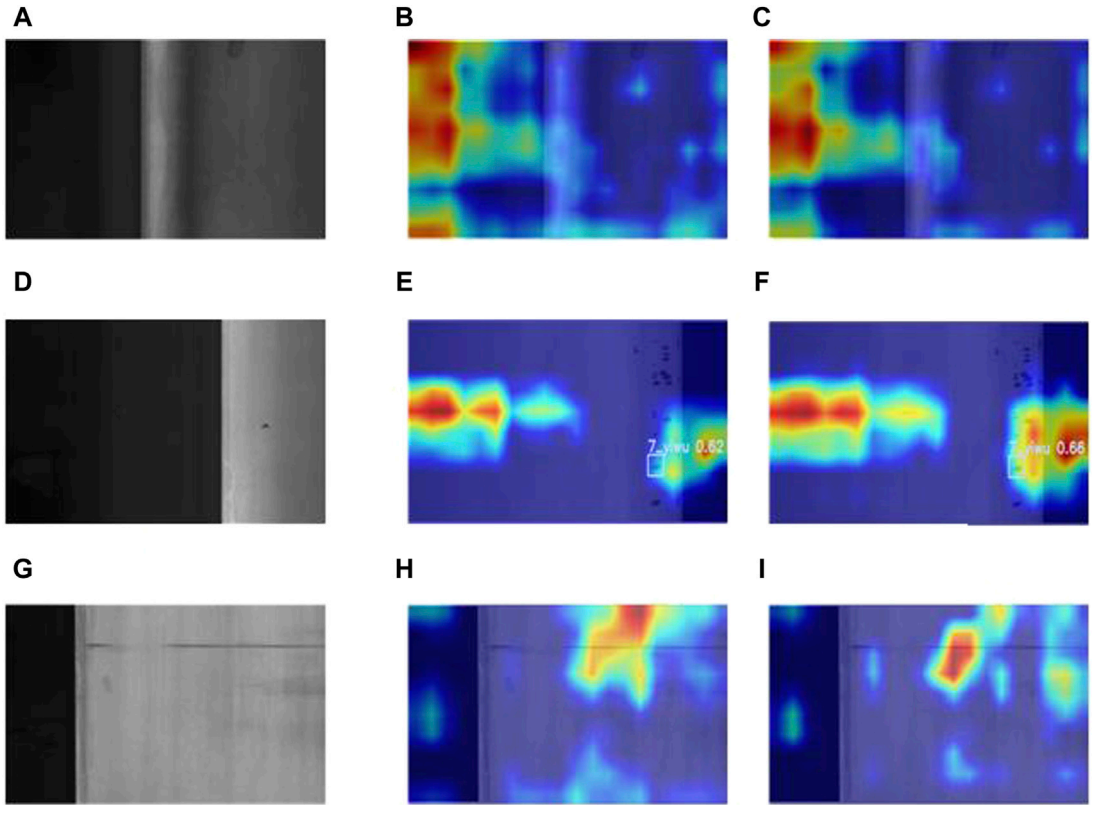



FIGURE 13
Thermal maps of some defect types in the dataset: **(A)** raw image of a steel oil spot,**(B)** heatmap of steel oil spot in YOLOv8, **(C)** heatmap of steel oil spots in PRS-YOLOv8, **(D)**original image of pitting defects, **(E)** thermal map of the pitted defect of the YOLOv8 model, **(F)** thermal map of pitted defects in PRS-YOLO, **(G)** raw image of steel inclusions, **(H)** thermal map of steel inclusions in YOLOv8, and **(I)** thermal map of steel inclusions in PRS-YOLOv8.

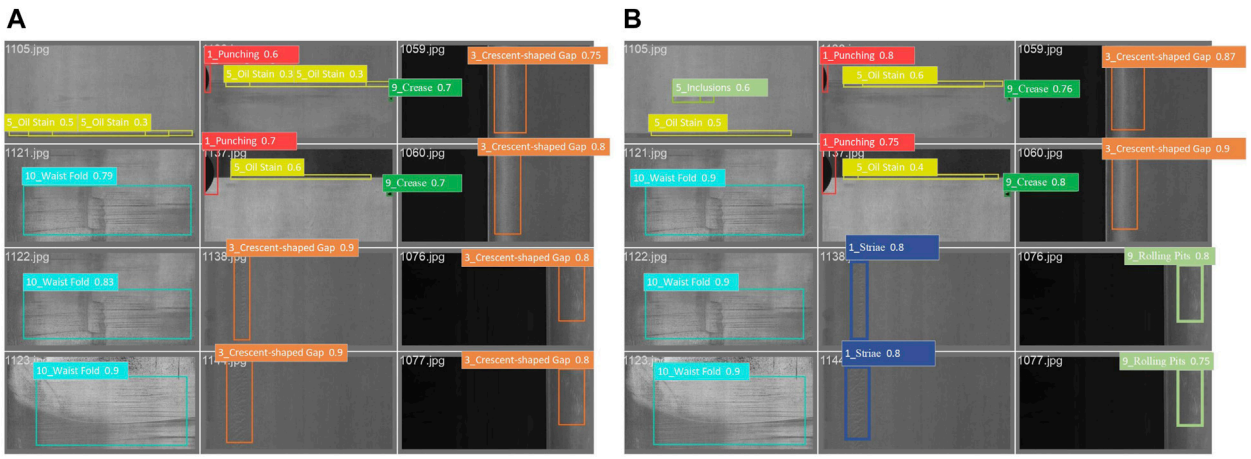



FIGURE 14
Comparison of test results: **(A)** YOLOv8 defect detection diagram and **(B)** PRS-YOLOv8 defect detection diagram.

including but not limited to punching, scratching, half-moon defects, etc. (see Figure 14). In order to ensure the fairness and comprehensiveness of the evaluation, the test dataset used in this paper is consistent, and the test dataset covers a variety of defect types, different sizes, complexities, and diverse background environments, aiming to simulate real industrial application scenarios and ensure the diversity and representativeness of the dataset. In this way, we validated the model's ability to detect not only pervasive defects, but also its accuracy in distinguishing highly similar defects (e.g., the missed detection problem of Inclusions, and the high false detection rate between Crescent-shaped Gap and Striae).

YOLOv8 encounters several challenges in the defect detection task of chemical special steels, especially when it comes to identifying defects containing inclusions, and it is easy to be confused in distinguishing defects with similar shapes (such as crescent-shaped gaps and stripes), revealing its limitations in dealing with defect categories with similar features. In contrast, the PRS-YOLOv8 model has significantly improved these problems through a series of innovative designs, not only greatly reducing the occurrence of missed and false detection events, but also showing excellent recognition accuracy when dealing with defect types that are difficult to accurately identify by YOLOv8. In addition, the PRS-YOLOv8 exhibits higher precision and positioning accuracy in the detection of all defect types, which greatly enhances the reliability and efficiency of the inspection results.

Overall, PRS-YOLOv8 has achieved significant progress in the field of defect detection compared to YOLOv8, showing stronger performance and accuracy both in small defect identification problems and in conventional defect detection.

### 4.4.7 Application deployment

In order to ensure the robustness and reliability of the PRS-YOLOv8 model in an actual industrial inspection system, we discuss several key factors in the model integration process, including hardware compatibility and strategies for handling changes in production line image acquisition conditions.

In real-world deployments, the hardware compatibility of the model is critical. Given the small number of parameters (only 3.29M) of the PRS-YOLOv8 model, this makes it easy to deploy on embedded devices such as industrial cameras and edge computing units. We chose a computing platform that supports the ARM architecture, ensuring that the model can run on a low-cost, low-power device while maintaining real-time processing power.

On the production line, changes in lighting conditions, camera position, and other factors can have an impact on inspection results. In order to alleviate the influence of these factors, we use adaptive histogram equalization technology to dynamically optimize the image contrast in image preprocessing, so as to improve the model's perception of the target defect area. This strategy effectively enhances the robustness of the model in complex environments, ensuring stable detection performance even under changing conditions.

## 5 Conclusion

In order to solve the problem that it is difficult to detect multi-category micro defects in chemical special steels, an enhanced YOLOv8 network architecture is proposed: PRS-YOLOv8. By introducing the adaptive histogram equalization technology, the algorithm dynamically optimizes the image contrast and improves the model's perception of the target defect area. The application of MPD-IOU loss function solves the problem of overfitting low-quality bounding boxes and improves the robustness of the model in complex scenarios. In addition, the addition of the ParC2Net module, the RepGFPN structure, and the RexSE-Head detection head effectively enhance the situational understanding and detection accuracy of the model, especially the capture of subtle features.

The experimental results show that compared with the most advanced small target detection algorithms Gold YOLO and EfficientDet-D0, PRS-YOLOv8 has excellent performance in small defect detection, with a score of mAP@0.5 as high as 93.5%, which significantly reduces the rate of missed detection and false alarm. In addition, the number of parameters of the model is only 3.29 MB, which is very suitable for resource-

constrained real-time application scenarios. However, the proposed method still has some limitations. Specifically, there is room for improvement in narrow defect detection, and the model's ability to generalize on unseen data or under different lighting conditions may be limited. Future work will focus on enhancing the detection ability of narrow defects by introducing strategies such as attention mechanism and serpentine convolution, and improving the adaptability of the model to diverse scenarios through transfer learning and increasing training data.

In summary, PRS-YOLOv8 has several key advantages over existing methods, including enhanced small target detection capabilities, good robustness to complex scenarios, and high efficiency and scalability. These advantages make it a promising solution for practical applications, while its limitations point the way for subsequent research and development.

## Data availability statement

The data analyzed in this study is subject to the following licenses/restrictions: Non-commercial. Requests to access these datasets should be directed to shaofeng yan yanshaofeng88888@gmail.com.

## Author contributions

YW: Resources, Writing–review and editing. SY: Conceptualization, Investigation, Methodology, Project administration, Software, Validation, Writing–original draft. HA: Data curation, Writing–original draft. SG: Funding acquisition, Writing–original draft. HZ: Funding acquisition, Writing–review and editing. XC: Investigation, Writing–original draft. HZ: Writing–review and editing, Supervision.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by Ministry of Education Humanities and Social Science Research Project (No. 23YJAZH034), the Postgraduate Research and Practice Innovation Program of Jiangsu Province (No. SJCX24_2147, SJCX24_2148), Enterprise Collaboration Project (No. Z421A22349, Z421A22304, Z421A210045).

## Acknowledgments

We thank the Ministry of Education Humanities and Social Science Research Fund, the Postgraduate Research and Practice Innovation Program of Jiangsu Province, Enterprise Collaboration Project for supporting this paper.

## Conflict of interest

Author HZ was employed by Ltd. Huai'an.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note



## References

1. Luo Q, Fang X, Liu L, Yang C, Sun Y. Automated visual defect detection for flat steel surface: © survey[J]. *IEEE Trans Instrumentation Meas*, 2020, 69(3): 626–44.doi:10.1109/tim.2019.2963555

2. Lin S, Ning S, Zhu H, Zhou T, Morris CL, Clayton S, et al. Neural network methods for radiation detectors and imaging. *Front Phys* (2023) 12: 1334298. doi:10.3389/fphy.2024.1334298

3. Duan J, Zhang H, Liu J, Gao M, Cheng C, Chen G. A dual-weighted polarization image fusion method based on quality assessment and attention mechanisms[J]. *Front Phys*, 2023, 11: 1214206, doi:10.3389/fphy.2023.1214206

4. Yang M, Lu S, Ding H, Chen J. Traffic safety assessment method of the immersed tunnel based on small target visual recognition image[J]. *Front Phys*, 2023, 11: 1159531, doi:10.3389/fphy.2023.1159531

5. Zhao W, Chen F, Huang H, Cheng W. A new steel defect detection algorithm based on deep learning[J]. *Comput Intelligence Neurosci*, 2021, 202:5592878–13.doi:10.1155/2021/5592878

6. He Y, Song K, Meng Q, Yan Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features[J]. *IEEE Trans instrumentation Meas*, 2019, 69(4): 1493–504.doi:10.1109/tim.2019.2915404

7. Boikov A, Payor V, Savelev R, Kolesnikov A. Synthetic data generation for steel defect detection and classification using deep learning[J]. *Symmetry*, 2021, 13(7): 1176, doi:10.3390/sym13071176

8. Mordia R, Verma AK. Visual technique for defects detection in steel products: a comparative study[J]. *Eng Fail Anal*, 2022, 134: 106047. doi:10.1016/j.engfailanal.2022.106047

9. Liang F, Zhou Y, Chen X, Liu F, Zhang C, Wu X. Review of target detection technology based on deep learning[C]. In: *Proceedings of the 5th international conference on control engineering and artificial intelligence* (2021). p. 132–5.

10. Zuo Y, Wang J, Song J. Application of YOLO object detection network in weld surface defect detection[C]//2021 IEEE 11th annual international conference on CYBER technology in automation, control, and intelligent systems (CYBER). IEEE (2021). p. 704–10.

11. Yang L, Wang Z, Gao S. Pipeline magnetic flux leakage image detection algorithm based on multiscale SSD network[J]. *IEEE Trans Ind Inform*, 2019, 16(1): 501–9.doi:10.1109/tii.2019.2926283

12. Hu K, Shen C, Wang T, Xu K, Xia Q, Xia M, et al. Overview of temporal action detection based on deep learning[J]. *Artif Intelligence Rev*, 2024, 57(2): 26, doi:10.1007/s10462-023-10650-w

13. Yang Y, Sun Q, Zhang D, Shao L, Song X, Li X. Improved method based on Faster R-CNN network optimization for small target surface defects detection of aluminum profile[C]. In: *2021 IEEE 15th international conference on electronic measurement and instruments (ICEMI)*. IEEE (2021). p. 465–70.

14. Wang H, Li M, Wan Z. Rail surface defect detection based on improved Mask RCNN [J]. *Comput Electr Eng*, 2022, 102: 108269, doi:10.1016/j.compeleceng.2022.108269

15. Shi X, Zhou S, Tai Y, Wang J, Wu S, Liu J, et al. An improved faster R-CNN for steel surface defect detection[C]. In: *2022 IEEE 24th international workshop on multimedia signal processing (MMSP)*. IEEE (2022). p. 1–5.

16. Xu S, Wang X, Lv W, Chang Q, Cui C, Deng K, et al. PP-YOLOE: an evolved version of YOLO[J]. *arXiv preprint arXiv:2203.16250*, 2022. doi:10.48550/arXiv.2203.16250

17. Wang C, He W, Nie Y, Guo J, Liu C, Wang Y, et al. Gold-YOLO: efficient object detector via gather-and-distribute mechanism[J]. *Adv Neural Inf Process Syst*, 2024, 36.

18. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers[C]. In: *European conference on computer vision*. Cham: Springer International Publishing (2020). p. 213–29.

19. Wang Y, Wang H, Xin Z. Efficient detection model of steel strip surface defects based on YOLO-V7[J]. *IEEE Access*, 2022, 10: 133936–44.doi:10.1109/access.2022.3230894

20. Akhyar F, Lin CY, Muchtar K, Wu TY, Ng HF. High efficient single-stage steel surface defect detection[C]. In: *2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE (2019). p. 1–4.

21. Xia K, Lv Z, Zhou C, Gu G, Zhao Z, Liu K, et al. Mixed receptive fields augmented YOLO with multipath spatial pyramid pooling for steel surface defect detection[J]. *Sensors*, 2023, 23(11): 5114, doi:10.3390/s23115114

22. Kou X, Liu S, Cheng K, Qian Y. Development of a YOLO-V3-based model for detecting defects on steel strip surface[J]. *Measurement*, 2021, 182: 109454, doi:10.1016/j.measurement.2021.109454

23. Jiang L, Yuan B, Wang Y, Ma Y, Du J, Wang F, et al. MA-YOLO: a method for detecting surface defects of aluminum profiles with attention guidance[J]. *IEEE Access*, 11, 71269, 86.doi:10.1109/access.2023.32915982023

24. Iqbal I, Shahzad G, Rafiq N, Mustafa G, Ma J. Deep learning-based automated detection of human knee joint's synovial fluid from magnetic resonance images with transfer learning [J]. *IET Image Process*, 2020, 14(10): 1990–8.doi:10.1049/iet-ipr.2019.1646

25. Iqbal I, Odesanmi GA, Wang J, Liu L. Comparative investigation of learning algorithms for image classification with small dataset[J]. *Appl Artif Intelligence*, 2021, 35(10): 697–716.doi:10.1080/08839514.2021.1922841

26. Zhu J, Zhou D, Lu R, Liu X, Wan D. C2DEM-YOLO: improved YOLOv8 for defect detection of photovoltaic cell modules in electroluminescence image[J]. *Nondestructive Test Eval*, 2024: 1–23.doi:10.1080/10589759.2024.2319263

27. Wang S, Wang Y, Chang Y, Zhao R, She Y. EBSE-YOLO: high precision recognition algorithm for small target foreign object detection[J]. *IEEE Access*, 11, 57951, 64.doi:10.1109/access.2023.32840622023

28. Stark JA. Adaptive image contrast enhancement using generalizations of histogram equalization[J]. *IEEE Trans Image Process*, 2000, 9(5): 889–96.doi:10.1109/83.841534

29. Su P, Han H, Liu M, Yang T, Liu S. MOD-YOLO: rethinking the YOLO architecture at the level of feature information and applying it to crack detection[J]. *Expert Syst Appl*, 2024, 237: 121346, doi:10.1016/j.eswa.2023.121346

30. Hu K, Li Y, Zhang S, Wu J, Gong S, Jiang S, et al. FedMMD: a federated weighting algorithm considering non-IID and local model deviation[J]. *Expert Syst Appl*, 2024, 237:121463, doi:10.1016/j.eswa.2023.121463

31. Roy AG, Navab N, Wachinger C. Concurrent spatial and channel 'squeeze and excitation'in fully convolutional networks[C]//Medical Image Computing and Computer Assisted Intervention–MICCAI 2018. In: *21st international conference, granada, Spain, september 16-20, 2018, proceedings, Part I*. Springer International Publishing (2018). p. 421–9.

32. Hu K, Zhang E, Xia M, Wang H, Ye X, Lin H. Cross-dimensional feature attention aggregation network for cloud and snow recognition of high satellite images[J]. *Neural Comput Appl*, 36, 2024: 7779–98.doi:10.1007/s00521-024-09477-5

33. Xie Y, Hu W, Xie S, He L. Surface defect detection algorithm based on feature-enhanced YOLO. *J Cogn Comput* (2019) 15(2):565–79. doi:10.1007/s12559-022-10061-z

34. Xu X, Jiang Y, Chen W, Huang Y, Zhang Y, Sun X. Damo-yolo: a report on real-time object detection design[J]. *arXiv preprint arXiv:2211.15444*, 2022. doi:10.48550/arXiv.2211.15444

35. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks[C]. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017). p. 1492–500.

36. Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design [C]. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021). p. 13713–22.

37. Siliang M, Yong X. Mpdiou: a loss for efficient and accurate bounding box regression. *arXiv preprint arXiv:2307.07662* (2023). doi:10.48550/arXiv.2307.07662

38. Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation [C]. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018). p. 8759–68.

39. Ge Z, Liu S, Wang F, Li Z, Sun J. *Yolox: exceeding yolo series in 2021[J]* (2021). arXiv preprint arXiv:2107.08430.

40. Tan M, Pang R, Efficientdet LQV. *Scalable and efficient object detection[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020). p. 10781–90.

frontiers | Frontiers in Medicine

TYPE Original Research
PUBLISHED 11 October 2024
DOI 10.3389/fmed.2024.1453405



OPEN ACCESS

EDITED BY
Zhenqiu Shu,
Kunming University of Science and Technology, China

REVIEWED BY
M. Badawy Abdel-Naser,
New York University, United States
Zhenyang Ding,
Tianjin University, China

*CORRESPONDENCE
Yong Guo
gy@fjpit.edu.cn
Zhifang Li
lizhifang@fjnu.edu.cn

RECEIVED 25 June 2024
ACCEPTED 16 September 2024
PUBLISHED 11 October 2024

CITATION
Zhang Z, Chen Z, Li Z, Zou J, Guo J, Chen K, Guo Y and Li Z (2024) Estimation of skin surface roughness *in vivo* based on optical coherence tomography combined with convolutional neural network.
*Front. Med.* 11:1453405.
doi: 10.3389/fmed.2024.1453405



# Estimation of skin surface roughness *in vivo* based on optical coherence tomography combined with convolutional neural network

Zhiqun Zhang[1], Zhida Chen[2], Zhenqian Li[2], Jian Zou[1], Jian Guo[1], Kaihong Chen[1], Yong Guo[1]* and Zhifang Li[1,2]*

[1]The Internet of Things and Artificial Intelligence College, Fujian Polytechnic of Information Technology, Fuzhou, Fujian, China, [2]Key Laboratory of Optoelectronic Science and Technology for Medicine, Ministry of Education, Fujian Provincial Key Laboratory of Photonics Technology, Fujian Provincial Engineering Technology Research Center of Photoelectric Sensing Application, College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou, Fujian, China

The texture of human skin is influenced by both external and internal factors, and changes in wrinkles can most directly reflect the state of the skin. Skin roughness is primarily used to quantify the wrinkle features of the skin. Therefore, effective and accurate quantification of skin roughness is essential in skincare, medical treatment, and product development. This study proposes a method for estimating the skin surface roughness using optical coherence tomography (OCT) combined with a convolutional neural network (CNN). The proposed algorithm is validated through a roughness standard plate. Then, the experimental results revealed that skin surface roughness including arithmetic mean roughness and depth of roughness depends on age and gender. The advantage of the proposed method based on OCT is that it can reduce the effect of the skin surface's natural curvature on roughness. In addition, the method is combined with the epidermal thickness and dermal attenuation coefficient for multi-parameter characterization of skin features. It could be seen as a potential tool for understanding the aging process and developing strategies to maintain and enhance skin health and appearance.

KEYWORDS

**skin roughness, optical coherence tomography, convolutional neural network, epidermal thickness, attenuation coefficient**

## 1 Introduction

With the global increase in the aging population, research on age-related alterations of skin is receiving growing interest (1). The passage of time and repeated exposure to UV radiation are the two main factors for aged skin. As age advances, there is a gradual loss of collagen in the skin, resulting in the development of wrinkles (2). Simultaneously, exposure to UV radiation can cause skin dryness, abnormal pigmentation, and other issues, ultimately leading to the formation of wrinkles on the skin (3). Quantifying skin wrinkles is of significant importance in the fields of skincare, medical treatment, and product development (4, 5).

The quantification of skin wrinkles allows for objective assessment of wrinkle severity, enabling accurate evaluation of treatment efficacy and product performance. Various methods

are used to quantify wrinkles, including both subjective and objective approaches. Subjective methods involve visual assessments by trained professionals or self-assessments by individuals themselves. These methods rely on scoring systems (five grades and nine grades) to evaluate the depth, length, and overall appearance of wrinkles (6, 7). However, subjective scoring relies more on individuals' subjective judgments and perceptions and often fails to capture minor changes.

In addition, objective methods utilize advanced imaging technologies and computer analysis to provide precise and quantitative measurements of wrinkle parameters. These methods can be divided into two-dimensional (2D) camera approaches and three-dimensional (3D) scanning techniques. Two-dimensional approaches for assessing skin include the use of mobile phone cameras with natural light sources (8), charge-coupled device (CCD) cameras utilizing UVA light sources (9), and speckles with laser light sources (10). However, two-dimensional photograph-based analyses by observers are vulnerable to noise, variable magnifications, and surrounding illumination. Furthermore, speckle contrast does not directly measure the height fluctuation of the skin surface. Three-dimensional scanning techniques contain 3D stereophotogrammetry (5) and phaseshift rapid *in vivo* measurement of the skin (PRIMOS) (11–13). However, motion artifacts during the image capture process in 3D stereophotogrammetry and PRIMOS can introduce errors, making it difficult to provide accurate and reliable measurements of skin roughness (14).

Optical coherence tomography (OCT) can overcome the above problems by providing non-invasive, real-time, and high-resolution imaging of the skin (15, 16). Surface roughness measurement based on OCT was proposed to assess the arithmetic mean roughness and average depth of roughness (17, 18). The roughness estimation was calculated based on the height relative to the central line of best fit through the dermal–epidermal junction (DEJ) (17). However, the central line of the skin surface differs from that defined by the International Organization for Standardization (ISO), which is based on the mean of height fluctuations (19). Additionally, image processing techniques such as the Gaussian filter, median filter, and differential filter were used to extract the ideal skin surface boundary (18). However, it is difficult for all skin since some empirical parameters in these image processing algorithms.

In this study, the method of OCT combined with the U-Net architecture of a convolutional neural network (CNN) is proposed for the evaluation of skin surface roughness using the advantages of 3D imaging and accurate boundary location. This choice is driven by the advantages of U-Net, namely, its ability to provide effective segmentation results and its limited requirement for training data. In this study, Section 2 introduces the OCT system, the accurate location of skin surface based on CNN, and the definition of arithmetic mean roughness and the depth of roughness. Section 3 first validates the algorithm using a roughness standard plate and explores the function of skin surface roughness in terms of age and gender. Section 4 offers a discussion of the findings and analyzes the strengths of the proposed methodology.

# 2 Materials and methods

## 2.1 Optical coherence tomography (OCT)

A schematic of our spectral domain optical coherence tomography (SD-OCT) system is illustrated in Figure 1A. The light source is a 12-mW superluminescent diode (SLD) with an FWHM bandwidth of 85 nm centered at 1310 nm (S5FC1021P, Thorlabs, Newton, NC, United States). Light is transmitted into a fiber coupler (FC) and then split into reference (50%) and sample (50%) arms, where collimators are used to obtain collimated light. A galvo scanning mirror (SM) and an achromatic lens (AL) with a focal length of 50 mm make up the scanning structure. The axial and lateral resolutions of the system in air are approximately 8.9 μm and 18.2 um, respectively. The detection arm consists of a spectrometer with a single line-scan camera (C-1235-1385, Wasatch Photonics, Logan, UT, United States) to construct a 3D image, resulting in the acquisition of 400 cross-sectional OCT images with a beam position increment of 25 μm.

A total of 16 volunteers were recruited for the experiment, including nine male individuals and seven female individuals. At the time of enrollment, subjects' ages ranged between 15 and 45 years, and all volunteers had no smoking history. Prior to the experiment, all volunteers signed an informed consent form, indicating their understanding and agreement to participate in the study. Before the imaging procedure, the region of interest of the skin was marked, washed using a cleansing cream, and exposed to a constant temperature and humidity in order to stabilize the experimental conditions. Subsequently, the volunteer was asked to place the back of the left hand on the designated area of the collection platform, as shown in Figure 1B, maintaining a fixed and comfortable posture. The collection platform was designed to support the hand and minimize any possible movement or vibration, ensuring the accuracy of data collection. Figures 1C,D show the typical cross-sectional and 3D OCT image of the back of the left hand. The texture of skin wrinkles is shown in Figure 1D. All the research procedures using human participants were carried out at Fujian Normal University with approval from the Institutional Review Board for the Protection of Human Subjects in Research (IRB).

## 2.2 Detecting boundary of skin surface using CNN

Figure 2 illustrates a flowchart of a CNN-based algorithm for detecting the boundary of the skin surface including boundary segmentation, curvature fitting, flattening, and boundary extraction, which will be described in detail in the following paragraph.

Before measuring skin roughness, it is necessary to segment the boundaries of the skin surface and flatten the skin surface. Figure 2 shows a CNN-based algorithm for detecting the real boundary of the skin surface. The skin surface was segmented and detected using a CNN (Figure 2B), specifically employing the U-Net architecture proposed by Ronneberger et al. (20), which has been widely used for biological image segmentation (21, 22). Meanwhile, ResNet50 was used as the backbone feature extraction network (23). The Adam optimizer was used to update the model, allowing the network to automatically adjust the learning rate for each parameter based on its update history (24). The learning rate (LR) for this experiment was set at 0.0001, which directly affected the speed and performance of the training process (25). A loss function of 0.01 quantified the error between actual values and predicted values (26). Mean Intersection over Union (MIoU) was used to evaluate the accuracy of the image segmentation model (27).

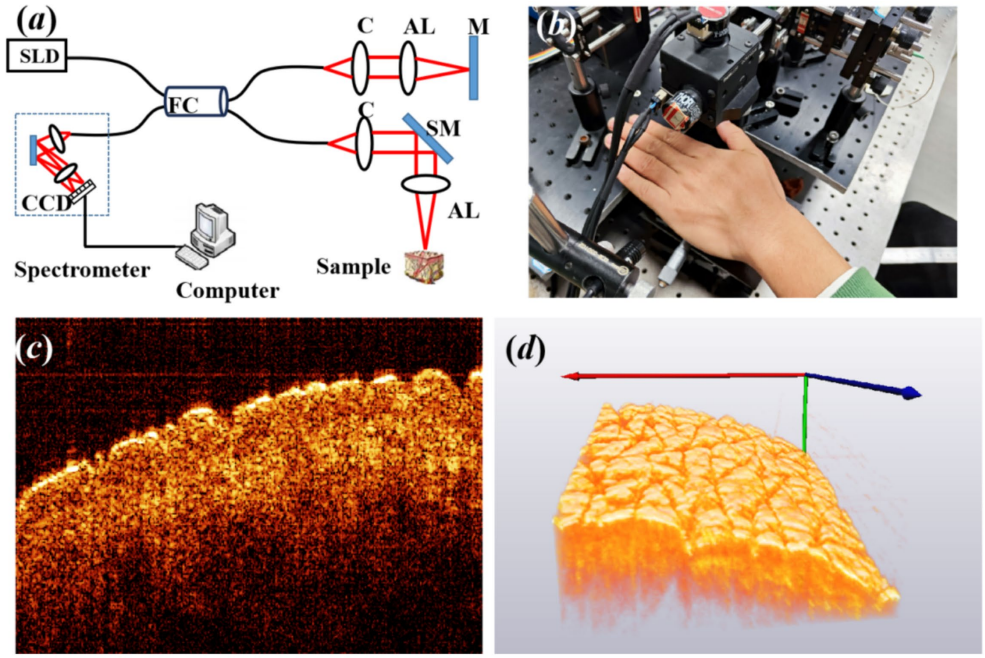



FIGURE 1
**(A)** Experimental setup of OCT, where SLD is the light source of the superluminescent diode, FC is fiber coupler, C is collimator, AL is achromatic lens, M is mirror, and SM is scanning mirror. **(B)** The back of the left hand for imaging, **(C)** typical cross-sectional OCT image, and **(D)** three-dimensional (3D) OCT image of the back of the left hand.

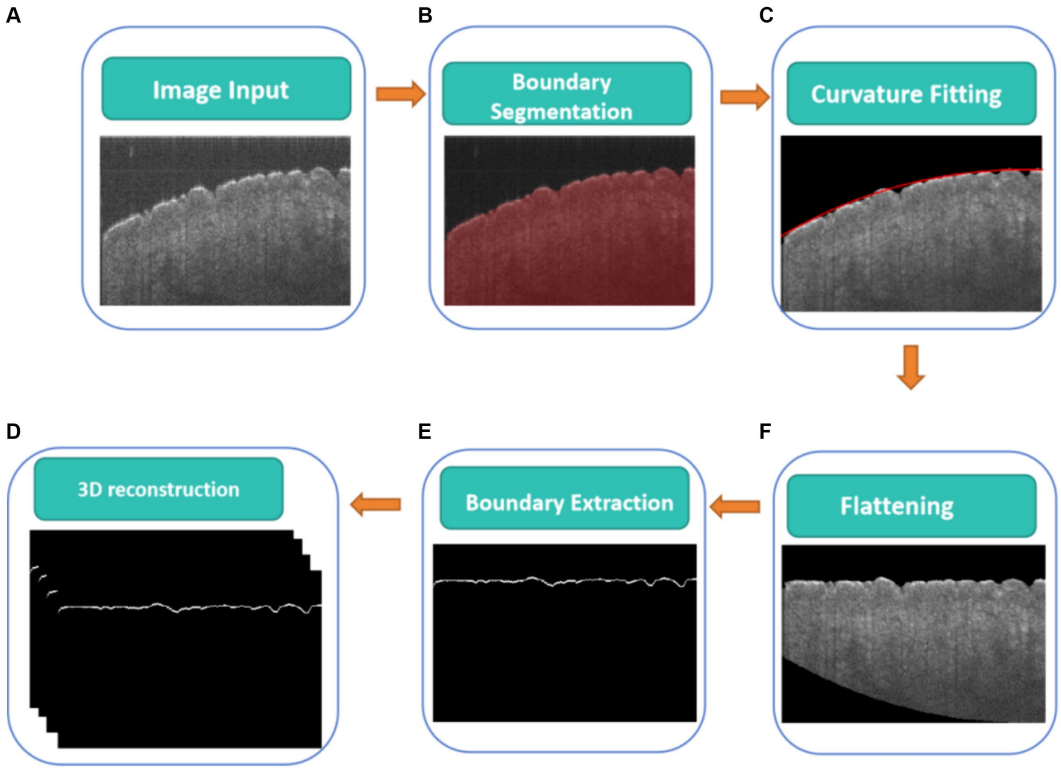



FIGURE 2
CNN-based algorithm for detecting boundary of skin surface, **(A)** original cross-sectional OCT images, **(B)** real boundary segmentation based on U-Net, **(C)** curvature fitting of real boundary height, **(D)** the flattening fitting boundary, **(E)** real boundary extraction on the flattening fitting boundary correction, **(F)** 3D real boundary.

In the experiment, a total of 16 sets of data were collected, amounting to 6,400 samples. Among these samples, 1,600 were annotated using Labelme for the boundaries of the skin. Afterward, the annotated dataset was typically divided into a training set and a test set in a 9:1 ratio. The training batch size was set at 8, and the number of iterations was set at 100. An MIoU score of 98.36 indicated a high degree of similarity between the model's predictions and the manual annotations, indicating a strong segmentation performance. In addition, Figure 3D shows that the noise in Figure 3A can be effectively reduced. It suggests that the model has successfully learned to extract the boundaries of the skin accurately, as shown in Figure 3, which lays a solid foundation for subsequent operations or tasks.

The boundary of the skin surface can be recorded based on the segmented image. However, the skin surface exhibits natural curvature, which can affect the assessment of roughness. Therefore, when calculating roughness, it is necessary to eliminate the influence of natural curvature. In this algorithm, the influence of natural curvature can be addressed by using the method of second-order polynomial fitting based on the least square method to find the curvature of the natural curvature in that region, as shown in Figure 2C. The flattening fitting boundary is shown in Figure 2D. Figure 4A demonstrates the fitting result of the skin. Subsequently, the curvature of the skin was flattened, as shown in Figure 4B, in which the fitting height of the boundary was set to the same height.

Once the acquisition of a cross-sectional skin boundary image was complete, the algorithm for 3D images of the skin surface was repeated to establish a three-dimensional (3D) topographic map of the skin, as shown in Figure 2F, and calculate 3D roughness data. Observations of the human skin surface under a stereomicroscope and OCT are shown in Figures 5A,B, respectively. Figure 5C shows a set of 400 B-scan images after segmenting the boundaries of the skin surface and flattening the skin surface. Figure 5D reveals the 3D reconstruction of Figure 5C, and the parameters of roughness were calculated based on Figure 5D. Figure 5A shows the skin roughness based on image texture, and Figure 5D shows the skin roughness according to the height, which is clearer than Figure 5A.

## 2.3 Quantification of surface roughness

According to the ISO 25178 standard established by the International Organization for Standardization (ISO), which is used for surface texture measurement, a series of surface texture parameters were defined to describe the morphology characteristics of a surface. Based on roughness standards and specific requirements, the arithmetic mean roughness ($R_a$) and the depth of roughness ($R_z$) were used for skin roughness. Their definitions are the arithmetic average of the absolute values of the surface height ($z$) and the maximum height between the highest peak and the lowest valley from the mean line within the measured region, respectively. The specific expressions of $R_a$ and $R_z$ are given as follows (19):

$$R_a = \frac{1}{n_x \times n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \left| z\left(x_i, y_i\right) \right|, \tag{1}$$

$$R_z = \max\left(z\right) - \min\left(z\right), \tag{2}$$

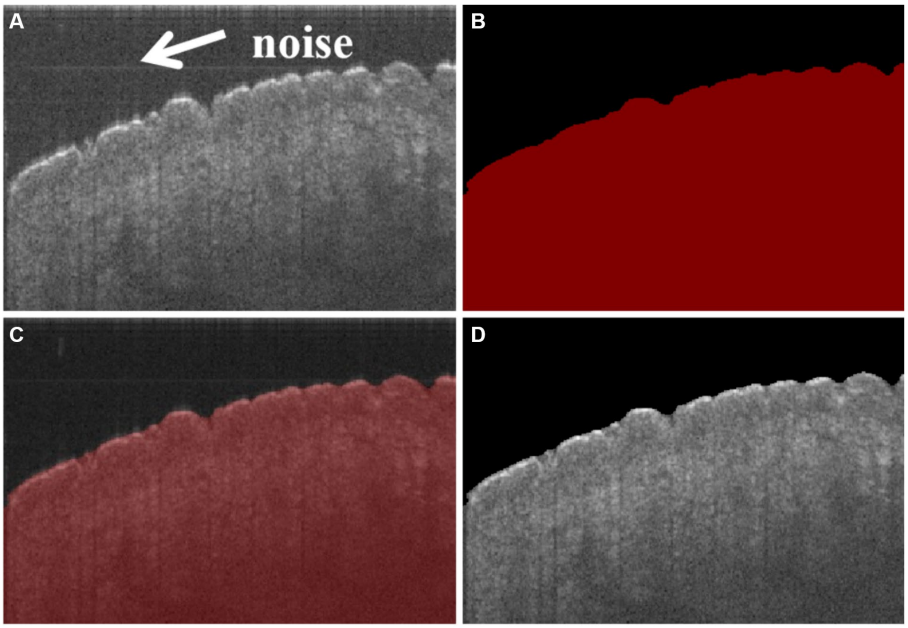



FIGURE 3
**(A)** Original cross-sectional OCT image of the skin, in which there is noise in the position of arrows, **(B)** masked image of skin segmentation based on CNN, **(C)** masked image superimposed with the original image, and **(D)** segmented image of the skin, in which the noise has been reduced comparing with **(A)**.

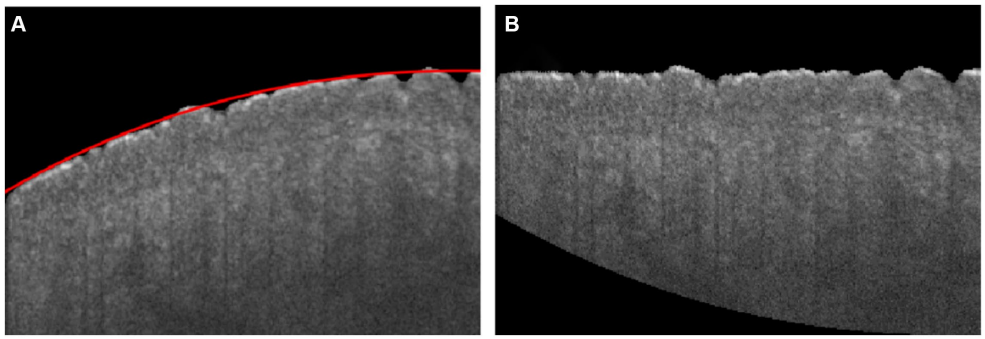



FIGURE 4
**(A)** Skin boundary curvature fitting, in which the red curve is the fitting boundary of skin; **(B)** Flattening boundary according to the fitting curve.

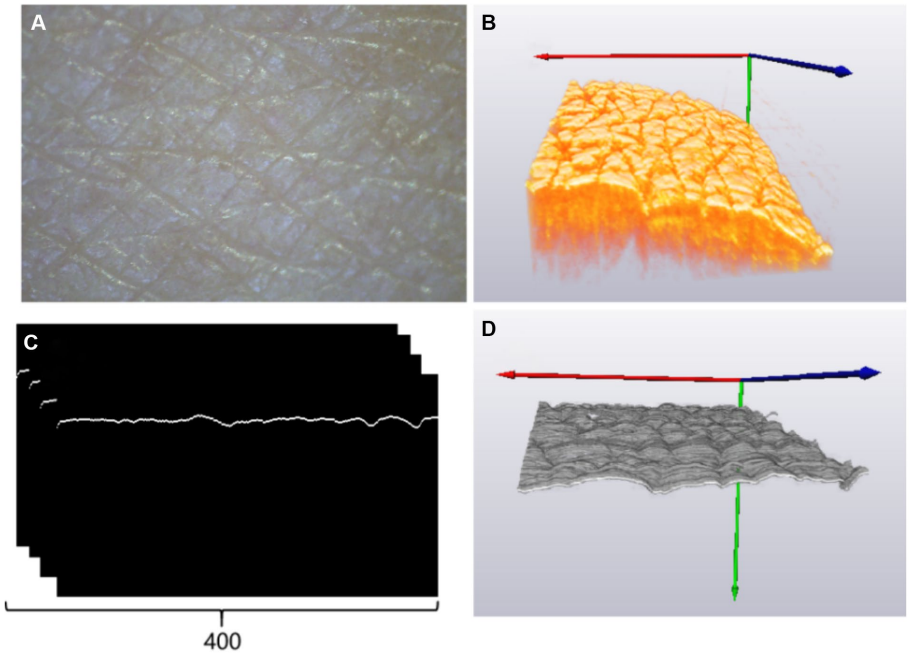



FIGURE 5
**(A)** Skin image of the back of the left hand under a stereomicroscope; **(B)** three-dimensional OCT image reconstruction results; **(C)** rear skin boundary; **(D)** three-dimensional skin boundary.

where $x_i$ and $y_i$ are two-dimensional spatial coordinates, respectively. Base on Equation 1, the arithmetic mean roughness ($R_a$) provides an overall measure of the surface roughness. Moreover, using Equation 2, the depth of roughness ($R_z$) indicates the maximum height variation on the surface. Both parameters including arithmetic mean roughness ($R_a$) and the depth of roughness ($R_z$) are related to the height fluctuation of the skin surface; thus, they depend on the axial resolution of OCT.

## 2.4 Statistical analysis

Correlation analysis was performed using Pearson's correlation coefficients. To test validity, the roughness parameters of $R_a$ and $R_z$ were compared to the age (Pearson's correlation). A Pearson correlation coefficient greater than 0.6 was considered a strong positive correlation.

# 3 Results

## 3.1 Validating the algorithm using a roughness standard plate

First, the proposed algorithm for skin roughness was validated using a roughness standard plate, which was purchased from Dongguan Tangxia Aiceyi Electronic Instrument Trading Company, as shown in Figure 6A. Figure 6A shows the roughness standard plate with an arithmetic mean roughness $R_a$ of 6.3 μm, which complies with the GB.T6060.2–2006 standard. Figure 6B indicates 3D OCT images

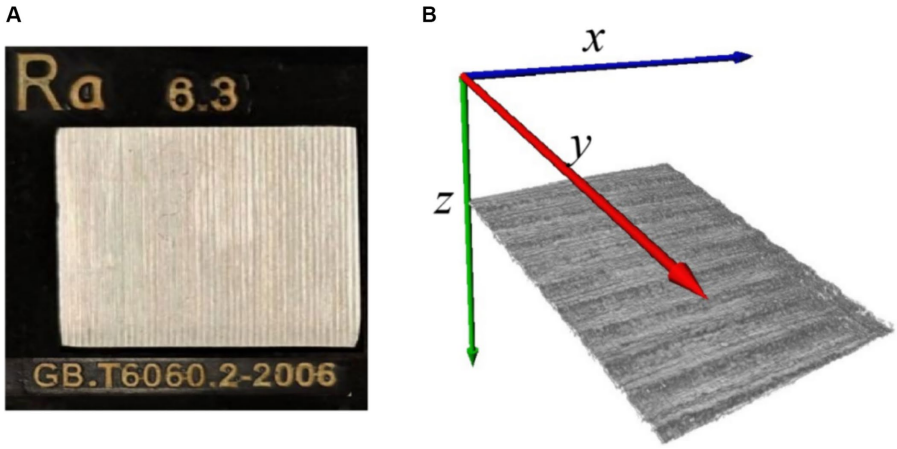



FIGURE 6
**(A)** Roughness standard plate; **(B)** 3D OCT image reconstruction of roughness standard plate.

TABLE 1 Calculated arithmetic mean roughness of the three positions based on the proposed method is consistent with the standard value in GB.T6060.2–2006.

| No. | Proposed method (μm) | Standard value (μm) | Error |
|---|---|---|---|
| 1 | 6.47 | | 2.7% |
| 2 | 6.17 | 6.3 | −2.1% |
| 3 | 6.39 | | 1.4% |

of the corresponding roughness standard plate, as shown in Figure 6A. Table 1 shows the arithmetic mean roughness $R_a$ on three positions of the roughness standard plate and demonstrates that the calculated value based on OCT is consistent with the standard defined in GB.T6060.2–2006. Thus, the proposed methods for roughness based on 3D OCT images provided an accurate and reliable measurement of roughness.

## 3.2 Skin surface roughness dependent on age

Figures 7A–C show the three-dimensional OCT images of the back of the hand's skin, illustrating how the skin surface flattens with age. The texture of the skin surface, as observed in these OCT images, depends on age. To quantify the texture, we utilized $R_a$ and $R_z$ to explore the function of the age based on the three-dimensional skin boundary images, as shown in Figures 7D–F. Higher $R_a$ values, shown in Figure 8A, indicate increased roughness, while higher values of $R_z$ in Figure 8B indicate deeper depths of roughness.

Figure 8A shows a significant positive correlation between age and the arithmetic mean roughness in which Pearson's correlation coefficients of men and women are 0.717 and 0.821, respectively. Meanwhile, there is a positive correlation between depth of roughness and age in Figure 8B, with Pearson's correlation coefficients of 0.626 and 0.833, respectively, for men and women. This can be attributed to the gradual loss of collagen, which leads to a decrease in elasticity and firmness in the skin. In addition, the slowing down of epidermal cell turnover is also a significant contributing factor to increased skin roughness (28, 29).

Figure 8 also demonstrates that the overall roughness levels, as indicated by the two parameters of arithmetic mean roughness $R_a$ and depth of roughness $R_z$, were higher in men than in women over the age of 25 years old because women generally place more emphasis on skincare compared to men (30, 31).

# 4 Discussion

The advantage of the proposed method in this study for estimating the roughness of skin surface is combined with other parameters such as epidermal thickness (32, 33) and dermal attenuation coefficient (17) based on OCT. Epidermal thickness was estimated based on the interval between the first peak and valley of the average OCT signal in terms of depth, and the attenuation coefficient was calculated based on the fitting line of the OC signal (Figure 9). Figure 10A reveals that the epidermal thickness is not correlated with age, which is consistent with the results found in the previous study (34).

In addition, as shown in Figure 10B, the attenuation coefficient of skin was found to be significantly decreased with increased age, which is consistent with a previous study (17). This is because of a gradual loss of collagen in the skin, resulting in an increase in roughness (2). The phenomenon was also observed in PS-SD-OCT, revealing depth-dependent correlations between the averaged dermal birefringence induced by collagen and the skin roughness parameters of the photoaged skin (35). The skin collagen would be determined using a two-photon confocal imaging for the skin surface (36). However, the image depth of a two-photon confocal image is lower than that of OCT.

Some studies employed traditional image processing techniques, including Gaussian filter, median filter, and differential filter, to emphasize the ideal surface boundary (18). However, these algorithms rely heavily on empirical values for different images. The proposed method in this study is accurate in extracting the surface boundary of skin to overcome the above problem since the CNN can effectively segment the skin surface (16, 22) through large-scale datasets and diverse data augmentation techniques for enhancing the generalization ability of models.

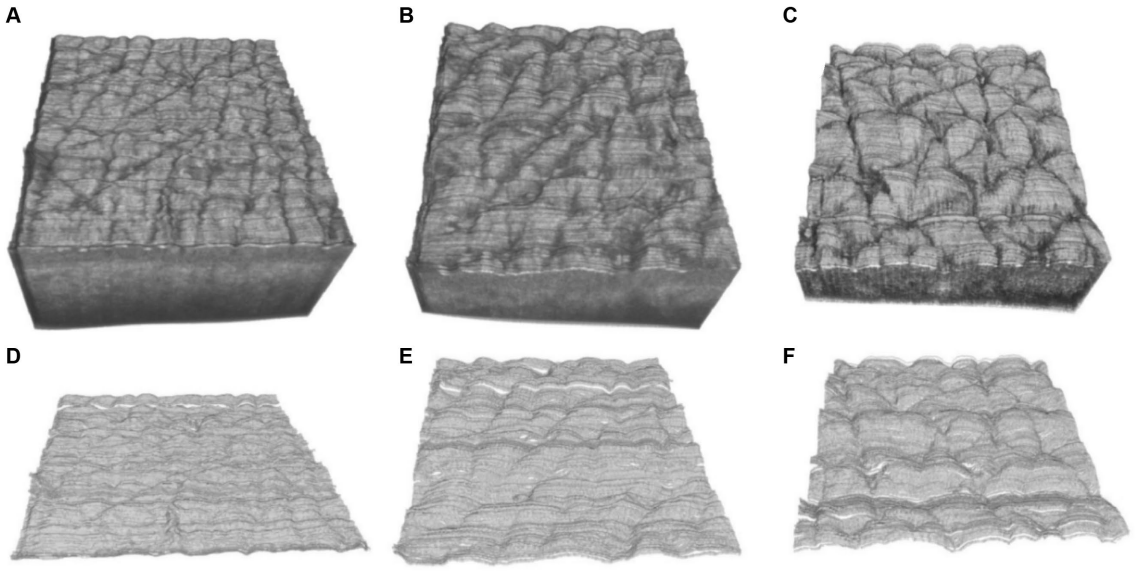



FIGURE 7
Three-dimensional OCT images at the ages of **(A)** 17, **(B)** 29, **(C)** 42 years, and **(D–F)** are the corresponding three-dimensional boundary images of **(A–C)**.

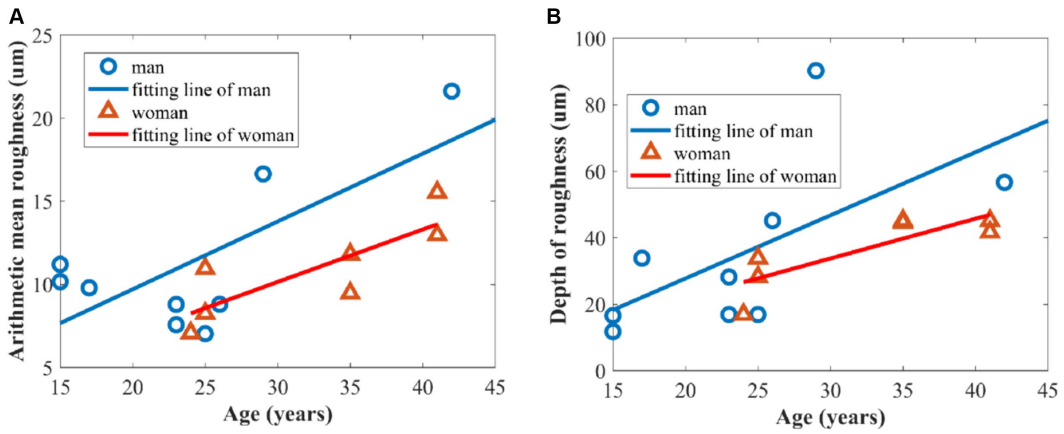



FIGURE 8
Relationship between skin roughness parameters and age of volunteers: **(A)** arithmetic mean roughness $R_a$; **(B)** depth of roughness $R_z$.

OCT directly measured the height fluctuation of the skin boundary for skin surface roughness, which was quantified by the arithmetic mean roughness and the depth of roughness. Thus, the development of OCT technology can improve the resolution of OCT, which, in turn, improves the accuracy of OCT image segmentation. In addition, the continuous progress in CNN algorithms further enhances the efficiency of the segmentation of skin boundaries.

## 5 Conclusion

In summary, the skin surface roughness is estimated using optical coherence tomography combined with CNN. The experimental results first demonstrated the effectiveness of the proposed algorithm by showing that the calculated value of the arithmetic mean roughness is consistent with the standard value for a roughness standard plate. In addition, the experimental results revealed that the skin surface roughness including the arithmetic mean roughness and depth of roughness depends on age and gender.

The advantage of the proposed method based on OCT is that it can reduce the effect of the skin surface's natural curvature on roughness and is combined with the epidermal thickness and dermal attenuation coefficient for multi-parameter characterization of skin features. Quantitative assessment of skin features including roughness, epidermal thickness, and attenuation coefficient enables researchers, clinicians, and cosmetic companies to monitor changes in skin condition over time, evaluate the effectiveness of interventions or treatments, and develop targeted products for anti-aging prevention. It serves as a valuable tool in understanding the aging process and

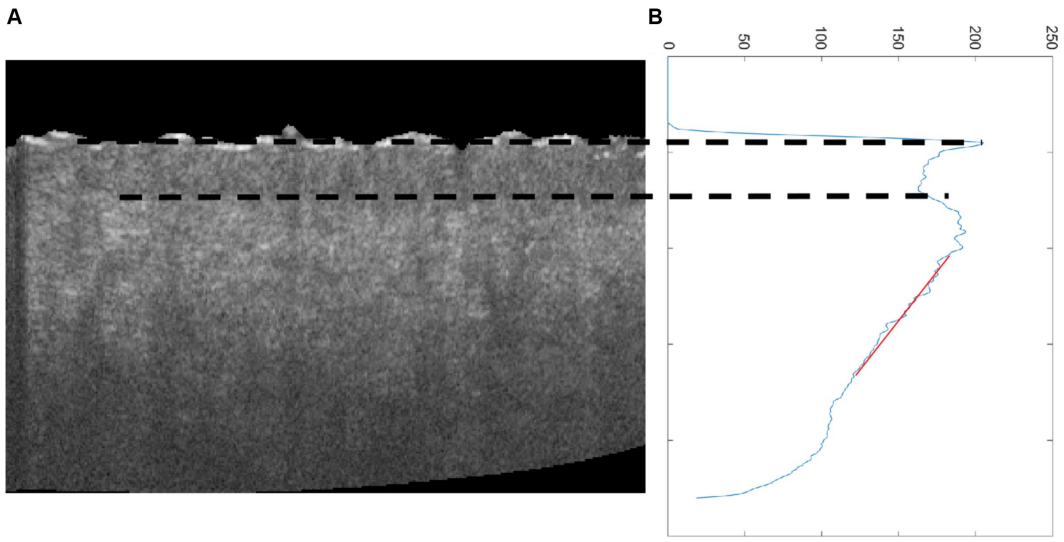



FIGURE 9
**(A)** Cross-sectional OCT image of skin and **(B)** average OCT signal dependent on depth; the two dot lines are the first peak and valley of average OCT signal, which denotes epidermal thickness, and the red line is the fitting line for estimating attenuation coefficient.

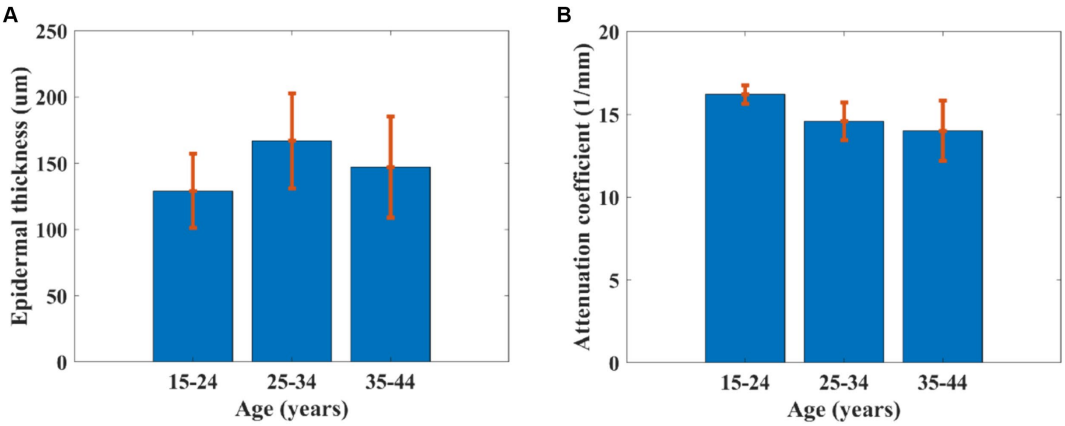



FIGURE 10
**(A)** Epidermal thickness and **(B)** attenuation coefficient of skin dependent on age.

developing strategies to maintain and enhance skin health and appearance.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

## Ethics statement

All the research procedures using human participants were carried out at Fujian Normal University in tight accordance with the Institutional Review Board for Protection of Human Subjects in Research (IRB) approval. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation in this study was provided by the participants' legal guardians/next of kin.

## Author contributions

ZZ: Conceptualization, Data curation, Investigation, Writing – original draft. ZC: Data curation, Formal analysis, Investigation, Writing – review & editing. ZQL: Data curation, Methodology, Resources, Software, Visualization, Writing – original draft. JZ: Formal analysis, Investigation, Validation, Writing – review & editing.

JG: Methodology, Software, Visualization, Writing – review & editing. KC: Investigation, Methodology, Writing – review & editing. YG: Formal analysis, Funding acquisition, Project administration, Writing – review & editing. ZFL: Conceptualization, Formal analysis, Investigation, Project administration, Supervision, Writing – review & editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded by the National Natural Science Foundation of China, grant number 61875038.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note



## References

1. Rittié L, Fisher L. Natural and Sun-induced aging of human skin. *Cold Spring Harb Perspect Med*. (2015) 5:a015370. doi: 10.1101/cshperspect.a015370

2. Varani J, Dame MK, Rittie L, Fligiel SEG, Kang S, Fisher GJ, et al. Decreased collagen production in chronologically aged skin: roles of age-dependent alteration in fibroblast function and defective mechanical stimulation. *Am J Pathol*. (2006) 168:1861–8. doi: 10.2353/ajpath.2006.051302

3. Krutmann J, Bouloc A, Sore G, Bernard BA, Passeron T. The skin aging Exposome. *J Dermatol Sci*. (2017) 85:152–61. doi: 10.1016/j.jdermsci.2016.09.015

4. Draelos ZD, Woodin FW. Clinical evidence of cell-targeted topical therapy for treating skin Dyspigmentation. *J Drugs Dermatol*. (2021) 20:865–7. doi: 10.36849/JDD.6037

5. Machado BHB, Silva IDDME, Pautrat WM, Frame J, Najlah M. Scientifc validation of three-dimensional Stereophotogrammetry compared to the IGAIS clinical scale for assessing wrinkles and scars after laser treatment. *Sci Rep*. (2021) 11:12385. doi: 10.1038/s41598-021-91922-9

6. Fujimura T, Sugata K, Haketa K, Hotta M. Roughness analysis of the skin as a secondary evaluation criterion in addition to visual scoring is sufficient to evaluate ethnic differences in wrinkles. *Int J Cosmet Sci*. (2009) 31:361–7. doi: 10.1111/j.1468-2494.2009.00521.x

7. Tsukahara K, Takema Y, Kazama H, Yorimoto Y, Fujimura T, Moriwaki S, et al. A photographic scale for the assessment of human facial wrinkles. *J Cosmetic Sci*. (2000) 2:186–94. doi: 10.1021/cc9900807

8. Bae JS, Lee SH, Choi KS, Kim JO. Robust skin-roughness estimation based on co-occurrence matrix. *J Vis Commun Image Represent*. (2017) 46:13–22. doi: 10.1016/j.jvcir.2017.03.003

9. Trojahna C, Dobos G, Schario M, Ludriksone L, Blume-Peytavi U, Kottner J. Relation between skin Micro-topography, roughness, and skin age. *Skin Res Technol*. (2014) 21:69–75. doi: 10.1111/srt.12158

10. Cheng C, Liu C, Zhang N, Jia T, Li R, Xu Z. Absolute measurement of roughness and lateral-correlation length of random surfaces by use of the simplified model of image-speckle contrast. *Appl Opt*. (2002) 41:4148–56. doi: 10.1364/AO.41.004148

11. Bloemen M. C. T., Martijn M.S.V.G., Wal M. B. A.Van Der, Verhaegen P. D. H. M., Middelkoop E. An objective device for measuring surface roughness of skin and scars. *J Am Acad Dermatol* (2011), 64, 706–715. doi: 10.1016/j.jaad.2010.03.006

12. Fujimura T, Haketa K, Hotta M, Kitahara T. Global and systematic demonstration for the practical usage of a direct in vivo measurement system to evaluate wrinkles. *Int J Cosmet Sci*. (2007) 29:423–36. doi: 10.1111/j.1468-2494.2007.00399.x

13. Fujimura T. Investigation of the relationship between wrinkle formation and deformation of the skin using three dimensional motion analysis. *Skin Res Technol*. (2012) 19:e318–24. doi: 10.1111/j.1600-0846.2012.00646.x

14. Jacobi U, Chen M, Frankowski G, Sinkgraven R, Hund M, Rzany B, et al. In vivo determination of skin surface topography using an optical 3D device. *Skin Res Technol*. (2004) 10:207–14. doi: 10.1111/j.1600-0846.2004.00075.x

15. Ulrich M., Themstrup L., Carvalho N.De, Manfredi M., Grana C., Ciardo S., et al. Dynamic optical coherence tomography in dermatology. *Dermatology* (2016), 232, 298–311. doi: 10.1159/000444706

16. Lin Y, Li D, Liu W, Zhong Z, Li Z. A measurement of epidermal thickness of fingertip skin from OCT images using convolutional neural network. *J Innov Opt Health Sci*. (2021) 14:2140005. doi: 10.1142/S1793545821400058

17. Vingan NR, Parsa S, Barillas J, Culver A, Kenkel JM. Evaluation and characterization of facial skin aging using optical coherence tomography. *Lasers Surg Med*. (2023) 55:22–34. doi: 10.1002/lsm.23611

18. Askaruly S, Ahn Y, Kim H, Vavilin A, Ban S, Kim PU, et al. Quantitative evaluation of skin surface roughness using optical coherence tomography in vivo. *IEEE J Selected Topics Quantum Electron*. (2018) 25:1–8. doi: 10.1109/JSTQE.2018.2873489

19. Amaral M. M., Raele M.P., Caly J. P., Samad R. E., Vieira N. D., Freitas A. Z. Roughness measurement methodology according to DIN 4768 using optical coherence tomography (OCT). in Proceedings of the Modeling Aspects in Optical Metrology II, Munich, Germany, (2009).

20. Ronneberger O., Fischer P., Brox T. U-net: convolutional networks for biomedical image segmentation. In The Proceeding of Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, (2015).

21. Mohammadi Z, Aghaei A, Moghaddam ME. CycleFormer: brain tissue segmentation in the presence of multiple sclerosis lesions and intensity non-uniformity artifact. *Biomed Signal Process Control*. (2024) 93:106153. doi: 10.1016/j.bspc.2024.106153

22. Kong C, Li D, Lin Y, Li Z. Automatic algorithm for the characterization of sweat ducts in a three-dimensional fingerprint. *Opt Express*. (2021) 29:30706–14. doi: 10.1364/oe.435908

23. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition, In The Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. Las Vegas, NV, USA, (2016).

24. Sun Y., Chen T., Yin W. An optimal stochastic compositional optimization method with applications to meta learning, In The Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3665–3669. Toronto, ON, Canada, (2021).

25. Gonzalez S., Miikkulainen R. Improved training speed, accuracy, and data utilization through loss function optimization, In The Proceeding of IEEE Congress on Evolutionary Computation, 1–8. Glasgow, UK, (2020).

26. Chaturvedi RP, Ghose U. A review of small object and movement detection based loss function and optimized technique. *J Intell Syst*. (2023) 32:20220324. doi: 10.1515/jisys-2022-0324

27. Everingham M, Eslami SMA, Gool LV, Williams CKI, Winn J, Zisserman A. The PASCAL visual object classes challenge: a retrospective. *Int J Comput Vis*. (2015) 111:98–136. doi: 10.1007/s11263-014-0733-5

28. Bocheva G, Slominski RM, Janjetovic Z, Kim T-K, Böhm M, Steinbrink K, et al. Protective role of melatonin and its metabolites in skin aging. *Int J Mol Sci*. (2022) 23:1238. doi: 10.3390/ijms23031238

29. Naughton GK, Jiang LI, Makino ET, Chung R, Nguyen A, Cheng T, et al. Targeting multiple hallmarks of skin aging: preclinical and clinical efficacy of a novel growth factor-based skin care serum. *Dermatol Ther*. (2023) 13:169–86. doi: 10.1007/s13555-022-00839-2

30. Mizukoshi K, Akamatsu H. The investigation of the skin characteristics of males focusing on gender differences, skin perception, and skin care habits. *Skin Res Technol*. (2013) 19:91–9. doi: 10.1111/srt.12012

31. Tsukahara K, Hotta M, Osanai O, Kawada H, Kitahara T, Takema Y. Gender-dependent Differences in Degree of Facial Wrinkles. *Skin Res Technol*. (2011) 19:e65–71. doi: 10.1111/j.1600-0846.2011.00609.x

32. Pezzini C, Ciardo S, Guida S, Kaleci S, Chester J, Casari A, et al. Skin ageing: clinical aspects and in vivo microscopic patterns observed with reflectance confocal microscopy and optical coherence tomography. *Exp Dermatol*. (2022) 32:348–58. doi: 10.1111/exd.14708

33. Ciardo S, Pezzini C, Guida S, Duca ED, Ungar J, Guttman-Yassky E, et al. A Plea for standardization of confocal microscopy and optical coherence tomography

parameters to evaluate physiological and Para-physiological skin conditions in cosmetic science. *Exp Dermatol*. (2021) 30:911–22. doi: 10.1111/exd.14359

34. Sandby-Møller J, Poulsen T, Wulf HC. Epidermal thickness at different body sites: relationship to age, gender, pigmentation, blood content, skin type and smoking habits. *Acta Dermato Venerol*. (2003) 83:410–3. doi: 10.1080/00015550310015419

35. Sakai S, Nakagawa N, Yamanari M, Miyazawa A, Yasuno Y, Matsumoto M. Relationship between dermal birefringence and the skin surface roughness of Photoaged human skin. *J Biomed Opt*. (2009) 14:044032. doi: 10.1117/1.3207142

36. Bachy M, Bosser C, Villain B, Aurégan J-C. Quantification of microstructural changes in the dermis of elderly women using morphometric indices of the skin surface. *Materials*. (2022) 15:8258. doi: 10.3390/ma15228258

frontiers | Frontiers in Physics

TYPE Review
PUBLISHED 18 October 2024
DOI 10.3389/fphy.2024.1485026



OPEN ACCESS

EDITED BY
Haoyu Chen,
University of Oulu, Finland

REVIEWED BY
Xianlu Tao,
Southeast University, China
Dai Jiangyan,
Weifang University, China

*CORRESPONDENCE
Wenhe Chen,
chenwh@jsut.edu.cn

RECEIVED 23 August 2024
ACCEPTED 02 October 2024
PUBLISHED 18 October 2024

CITATION
He W, Chen W, Tian S and Zhang L (2024) Towards full autonomous driving: challenges and frontiers.
*Front. Phys.* 12:1485026.
doi: 10.3389/fphy.2024.1485026



# Towards full autonomous driving: challenges and frontiers

Wei He[1], Wenhe Chen[2,3]*, Siyi Tian[4] and Lunning Zhang[5]

[1]Shanghai Engineering Research Center of AI & Robotics, Academy for Engineering and Technology, Fudan University, Shanghai, China, [2]Artificial Intelligence Industry Academy School of Computer Engineering, Jiangsu University of Technology, Changzhou, China, [3]Shanghai Huace Navigation Technology Co., Ltd., Shanghai, China, [4]School of Sensing Science and Engineering, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, [5]Shanghai Future Space-Time Technology Co., Ltd., Shanghai, China

With the rapid advancement of information technology and intelligent systems, autonomous driving has garnered significant attention and research in recent years. Key technologies, such as Simultaneous Localization and Mapping (SLAM), Perception and Localization, and Scene Segmentation, have proven to be essential in this field. These technologies not only evolve independently, each with its own research focus and application paths, but also complement and rely on one another in various complex autonomous driving scenarios. This paper provides a comprehensive review of the development and current state of these technologies, along with a forecast of their future trends.

KEYWORDS

**autonomous driving, simultaneous localization and mapping, perception and localization, scene segmentation, deep learning**

## 1 Introduction

Autonomous driving has developed rapidly in the past 2 decades and is now gradually evolving towards full automation. The premise for autonomous vehicles to achieve high-level tasks such as decision-making and planning is to obtain accurate self-state and environmental perception information in various complex scenarios, among which technologies such as Simultaneous Localization and Mapping (SLAM), Perception and Localization, Point Cloud Completion and Scene Segmentation are crucial, as shown in Figure 1. Specifically, SLAM is the basic framework for information association between agents and the environment, which provides agents with the ability to construct and locate real-time environment maps. Agents need to interact with the environment with a high degree of autonomy, and the Perception and Localization technology of autonomous driving systems is particularly critical. It covers a series of advanced functions from environmental perception to precise autonomous positioning. Scene Segmentation greatly enhances the agent's understanding and adaptability to complex scenes by performing detailed semantic analysis of the environment.

This paper will detail the development history, current implementation mechanisms and their practical roles in autonomous driving and broader computer vision and 3D data processing of these key technologies. Through in-depth analysis of the current situation and challenges of these technologies, this paper aims to explore their development trends and forecast how to improve the efficiency and intelligence level of the overall system through technology integration. In addition, it will also predict the future development direction of these technologies and their potential role in promoting the Frontier of automation and intelligent technologies.

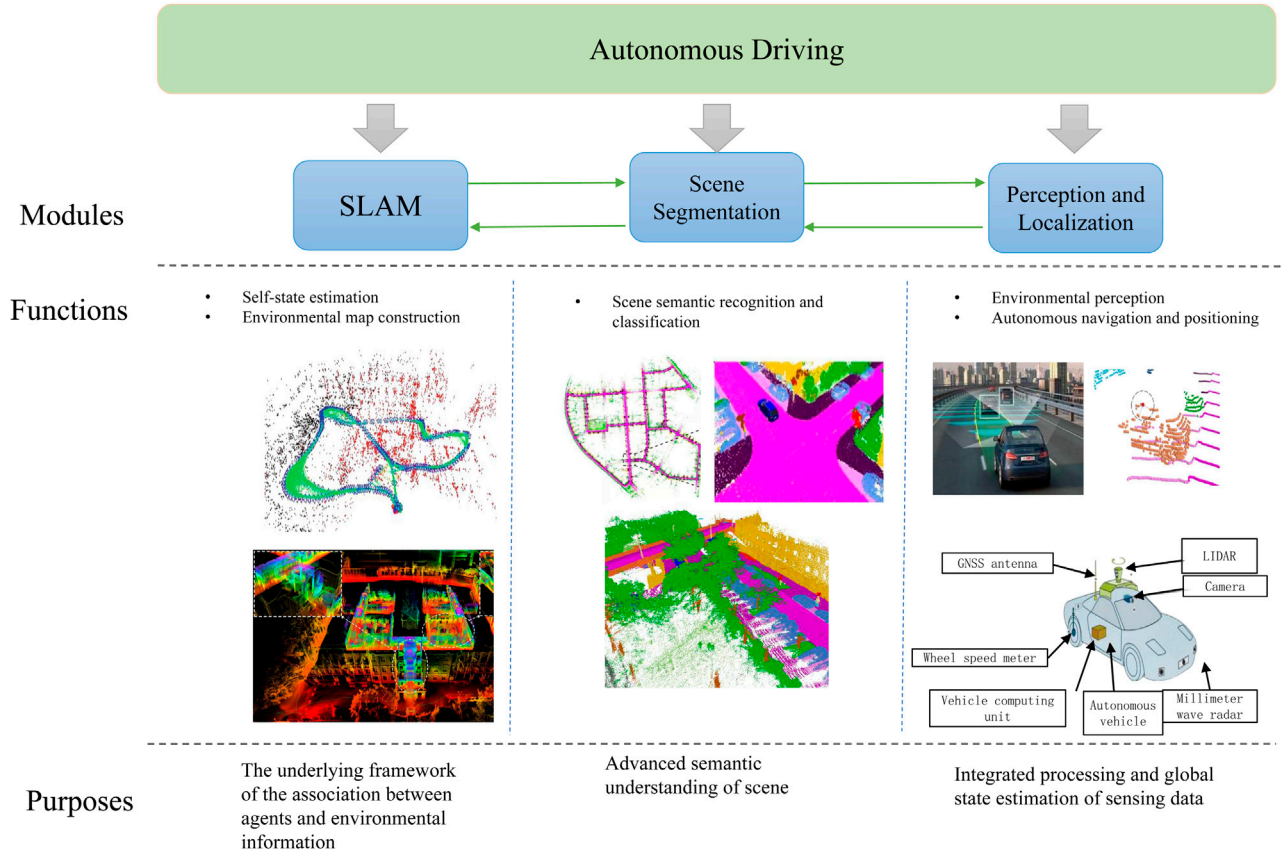



FIGURE 1
The overall framework of the paper.

# 2 Simultaneous Localization and Mapping (SLAM)

## 2.1 Definition, basic principles and development history of SLAM

Simultaneous Localization and Mapping (SLAM) is a technology in which a robot estimates its own state (position, speed, direction, sensor bias, etc.) in an unknown environment, and simultaneously constructs its motion environment based on sensor perception information. Over the past 30 years, there were many significant progress made in SLAM field which has been widely used in many industries. The basic principles and development of visual SLAM, laser SLAM and multi-sensor fusion SLAM in the order of different main sensors will be introduced in this section. The specific development route is shown in Figure 2. Due to the widespread application of the fusion of Inertial Measurement Unit (IMU) and SLAM, the development of such applications is also described in 2.1.1 and 2.1.2.

### 2.1.1 Visual SLAM

In the early stage of visual SLAM research, most of them belong to filtering-based methods, such as EKF-based MonoSLAM [1], tight coupling system composed of IMU and monocular camera [2] which realize real-time operation for the first time. Mourikis et al. [3] proposed the famous MSCKF based on the conventional EKF (Extended Kalman Filter). The state vector of MSCKF contains multiple camera states, and the measurement of the same feature point is used to define constraints between two or more camera poses. When some specific conditions are met, these constraints are used for filter updates. Compared with the conventional EKF method, the advantage of the MSCKF method is to maintain only one state variable with low dimension, and no longer store the coordinate information of map points, so as to reduce the amount of storage and calculation. MSCKF algorithm has become one of the classic algorithms of VIO, but it does not optimize the location of map points in the scene, therefore it is difficult to ensure the overall positioning accuracy for a long time. Optimization-based SLAM method is another mainstream solution, which optimizes the robot pose to be solved and the position of spatial waymark points through Bundle Adjustment (BA) technology. Compared with filtering-based methods, optimization-based methods usually achieve stronger robustness and higher accuracy, and their framework is more flexible. But it is more computational and time-consuming because its multiple iterative optimization process requires more computing resources. In 2007, Klein et al. [4] proposed the famous PTAM (Parallel Tracking and Mapping) algorithm, which applied graph optimization theory to solve SLAM problems for the first time, meanwhile, this algorithm pioneered the parallel implementation of

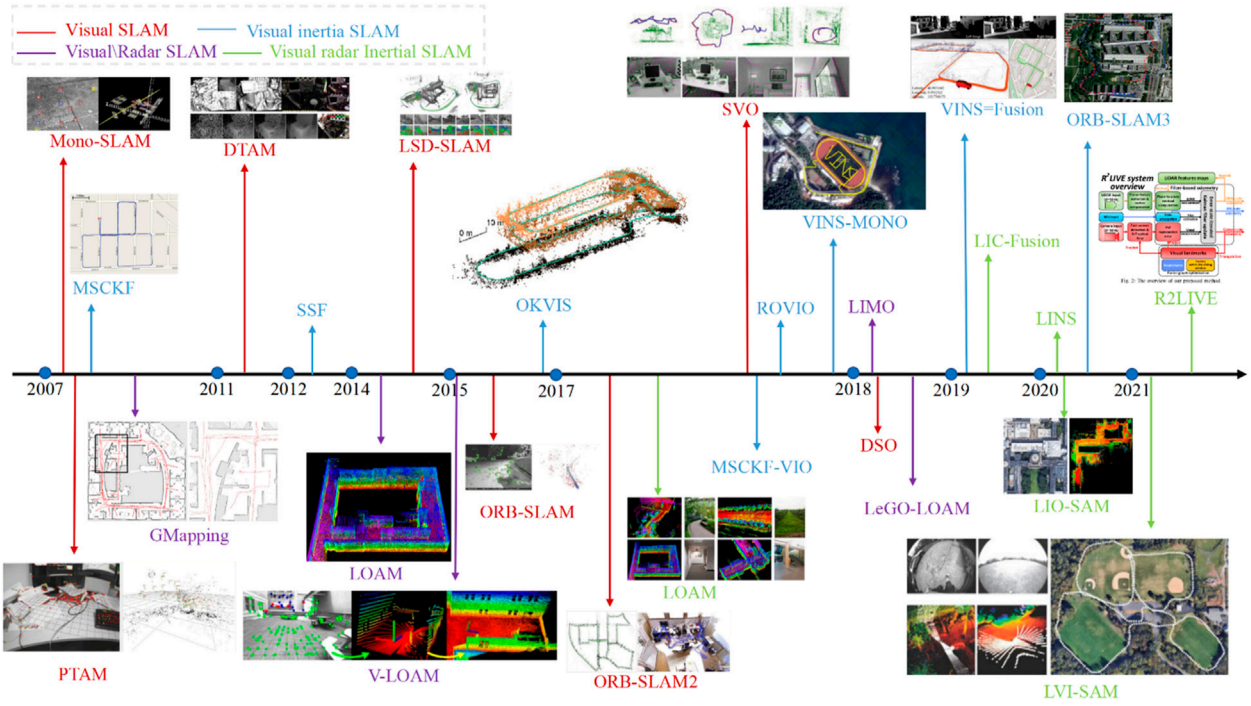



FIGURE 2
SLAM technology development roadmap.

locating and mapping on two independently running threads. Mur-Artalet [5] continued and improved the basic idea of PTAM, and the famous ORB-SLAM algorithm was proposed. ORB-SLAM is a more complete monocular SLAM system, which includes three threads: tracking, partial mapping and loopback detection. In each thread, ORB (Oriented FAST and Rotated BRIEF) operator [6] is used to extract and describe image features. In 2017, Mur-Artal [7] further proposed ORB-SLAM2, which supports both monocular camera input and binocular cameras and RGB-D cameras input. Similar to the filtering-based method, the combination of visual inertial sensors is often used to build visual inertial SLAM system in the optimization-based framework. Leutenegger [8] proposed keyframe-based OKVIS (Open Keyframe-based Visual-Inertial Syetem). In 2021, Campos et al. [9] proposed ORB-SLAM3, which can support monocular, binocular, and RGB-D image input of pinhole or fisheye lens models, and can perform visual, visual inertial, and multi-map SLAM processes. VINS-Mono (Monocular Visual-Inertial Navigation System) proposed by Qin et al. [10] is another representative work of optimization methods. VINS-Mono uses a sparse direct method similar to SVO [11] as its front end, simple corner points are extracted on the image, and the corner points are tracked by KLT (Kanade-Lucas-Tomasi) optical flow method. In constructing the BA, a quaternion-based IMU pre-integration model [12], a sliding window, and a two-step marginalization technique are used. Depending on the type of visual front end, all of the above visual SLAM methods can be referred to as feature point methods. Feature point method has long been regarded as the mainstream method of SLAM, but the disadvantage of this method is that it easily leads to poor feature extraction or feature tracking effect when encountering weak texture environment, fast robot movement speed or blurred visual imaging, which affects the performance of the algorithm, and the key point extraction and descriptor calculation are also time-consuming. Therefore, some researchers have also studied another kind of direct SLAM method, which directly estimates camera motion according to pixel gray information. Concha [13] proposed a monocular visual inertial odometer VIDS (Visual-Inertial Direct SLAM) based on the direct method. Forster proposed a SVO (Semi-Direct Monocular Visual Odometry), which combines feature point method with direct method. SVO executes motion estimation thread and map construction thread in parallel, and can obtain fast and accurate positioning effect when the observation scene is approximately plane. Engel et al. [14] proposed a monocular SLAM method LSD-SLAM (Large-Scale Direct Monocular SLAM) based on direct method, which can obtain more accurate motion estimation in large-scale scenarios and construct large-scale environmental maps. On this basis, Engel et al. [15] incorporated the photometric calibration strategy and further exploited sparsity, and proposed the DSO (Direct Sparse Odometry) algorithm. Stumberg [16] proposed the VI-DSO method with further fusion of IMU measurement information.

In addition, thanks to the development of semantic segmentation technology and object detection technology based on deep learning, the integration of higher-level semantic information into the design and implementation of SLAM algorithm has become a new direction for researchers. Bowman et al. [17] used probabilistic representation to theoretically analyze the solution of semantic SLAM problems, and proposed a theoretical framework for semantic data association and iterative solution in SLAM by using Expectation-Maximum algorithm (EM). On this

basis, Lianos et al. [18] proposed a semantic SLAM algorithm VSO (Visual Semantic Odometry) that uses semantic information to assist visual feature tracking. Yang et al. [19] proposed a monocular SLAM algorithm that fuses indoor plane features (walls, floors, etc.) with object-level road signs. Frost et al. [20] solved the problem of missing scale in monocular SLAM by constructing 2D projection constraints of vehicle targets with known scales in BA. Nicholson et al. [21] proposed a three-dimensional modeling method of object-level road marks, i.e., an ellipsoid is used to represent three-dimensional object road marks, and a semantic constraint residual term with geometric significance is added to the optimization function of BA to improve positioning accuracy. Li et al. [22] proposed that the closed-loop detection function in complex situations such as large viewing angle changes and occlusion can be enhanced by constructing object-level semantic mapping.

### 2.1.2 Radar SLAM

The measurement data of LiDAR is a point cloud, and each point cloud contains the spatial coordinates of many spatial points in the Ontology coordinate system at the time of LiDAR measurement. LiDAR SLAM uses point cloud registration, i.e., pose estimation is realized by finding the matching item between the source frame and the target frame and inferring the pose transformation from the source frame to the target frame.

Early LiDAR SLAM studies have mainly focused on 2D LiDAR, and several 2D laser SLAM based on filtering and optimization frameworks have been proposed, including EKF-based frameworks, Unscented Kalman Filter (UKF)-based frameworks [23], and classic framework GMapping [24] based on Particle Filter (PF) [25]. A representative work of graph optimization-based methods is GraphSLAM [26].

With the development of technology, SLAM based on 3D LiDAR has gradually become a research hotspot. The research focus of SLAM method based on 3D LiDAR is mainly on point cloud registration because the basic theory of SLAM has gradually matured when 3D LiDAR began to be studied. Iterative Closest Points (ICP) [27] is the most classic point cloud registration method, which correlates points in a source frame with points in a target frame according to the nearest neighbor criterion, and then solves the optimal transformation between two point clouds. Based on ICP, Mendes et al. [28] proposed to achieve positioning by ICP registration between the current frame and key frames, and then detect loopbacks by ICP registration between different key frames. In order to overcome the defects that the original ICP is sensitive to initial values and measurement noise, many variants of ICP were proposed and applied to LiDAR SLAM. According to the curvature, LOAM [29] extracts surface feature points and corner feature points from the point cloud, and these feature points are registered with adjacent frames and world maps through point-surface and point-line ICP to realize low drift pose estimation. Based on LOAM, LeGO-LOAM [30] introduces ground point constraints in inter-frame registration to suppress height drift, and the pitch angle, roll angle and vertical axis coordinates related to height are first optimized, and then other pose components are optimized, which improves the solution efficiency of inter-frame registration. Also based on point-surface ICP, IMLS-SLAM [31] and SuMa [32] represent planes in maps in the form of hidden planes and patches, respectively. ICP based on normal distribution describes the local geometry of the point cloud through the local covariance matrix of the point cloud, so that the registration takes into account the local orientation of the point cloud. Among them, the representative methods are Normal Distribution Transformation (NDT) [33] and Generalized ICP (GICP) [34].

In addition to ICP, researchers are also actively exploring the application of other point cloud registration schemes in LiDAR SLAM. S4-SLAM [35] uses Super4PCSI [36], a method for point cloud registration based on affine invariance of line segment crossover ratio. GP-SLAM+ [37] uses Gaussian process regression to predict "test points" evenly distributed in space on the current point cloud, and then registers them with the results predicted from the map. SegMap [38] uses machine learning to extract feature points and calculate descriptors from the point cloud, adds semantic information to the point cloud, which can achieve more robust registration, and can reach a pose output frequency of 1 Hz, so as to lay a foundation for the introduction of subsequent machine learning methods.

### 2.1.3 Multi-sensor fusion SLAM

Generally, multi-sensor fusion positioning methods can be divided into loose coupling method and tight coupling method. The former fuses the independent positioning results of single sensors, while the latter fuses the original measurement information of various sensors.

As the cost of sensors decreases, SLAM methods that integrate three or more sensors have attracted more and more attention from academia and industry in order to obtain higher precision and robust performance and further extend the applicable scenarios of SLAM systems. In 2018, Zhang et al. [39] proposed a sequential multi-sensor fusion SLAM-VLOAM. In this method, IMU firstly provides pose prediction for a loosely coupled VIO, and then the localization results of the VIO are further loosely coupled with LiDAR data to realize a pose estimation from coarse to fine. LVI-SAM [40] combines the VIO system and the LIO system to construct a tightly coupled LVIO. Among them, VIO provides the initial value for the point cloud registration of LIO, and the output of LIO system helps the VIO system to initialize and obtain the depth of visual feature points. Moreover, LVI-SAM also detects the working conditions of these two subsystems respectively. When one subsystem fails, the other system can run independently to ensure the robustness of the system. At the back end, LVI-SAM uses a factor map to receive the inter-frame pose constraints provided by the two subsystems to smooth the trajectory and improve the overall estimation accuracy. Based on FAST-LIO and VINS-Mono, R2LIVE [41] uses ESIKF to tightly couple IMU data with camera data and LiDAR data respectively, and uses a local factor map to adjust key frame pose and visual feature point position. LIC-Fusion [42] is based on the architecture of tightly coupled VIO method MSCKF, LiDAR frames are introduced on the basis of visual frames, and the constraints of LiDAR common view features are added between LiDAR frames. Meanwhile, the external parameters and time differences between sensors are estimated as filtering parameters, which achieves tight coupling well. LIC-Fusion2.0 [43] proposes a more robust plane tracking method between LiDAR frames on the basis of LIC-Fusion, which further improves the system performance.

## 2.2 Application cases of SLAM in different fields

Since SLAM is essentially autonomous positioning and environmental information correlation in unknown environments, and involves a variety of sensors, direct needs exist in many fields. Therefore, the application of SLAM technology in various industries has been fully studied after decades of development, covering robotics, industrial automation, autonomous driving, augmented reality, medical care, aerospace, geology and environmental science, military security, etc.

Robotics is the hottest field of SLAM technology application. In indoor environments, service robots use SLAM for localization mapping to autonomously navigate and perform tasks in hotel, hospital and home environments [44]; SLAM is used for autonomous navigation and intelligent obstacle avoidance of material delivery trolleys on the factory floor to improve logistics efficiency and automation levels [45]; Unmanned aerial vehicles can use SLAM to carry out autonomous flight [46], and realize surveying and mapping, express delivery and other tasks. Autonomous driving vehicles rely on SLAM to build high-precision maps and assist vehicles in path planning, obstacle avoidance and positioning to ensure driving safety [47]. On AR and VR, SLAM enables such devices to build and update virtual environment maps in real time [48], and can be further used for highly immersive gaming experiences, create dynamic and interactive learning environments or help designers create virtual prototypes and simulations in the field of industrial design; In the medical field, SLAM can also be used for surgical navigation, assisting the safe movement of instruments by building a high-precision model of the surgical area. In the military field, SLAM helps reconnaissance drones navigate and position under the denial condition of no external available signals, and realize tasks such as reconnaissance, surveillance, and target tracking [49].

## 2.3 Key issues and challenges of SLAM technology

### 2.3.1 Front-end data association

The SLAM front-end module is responsible for feature extraction, description and tracking on the raw measurement data of the sensor, so as to establish data association on continuous time frames. The state of the carrier can be preliminarily estimated and optimized based on the correctly associated image or point cloud frame. The results of front-end estimation are crucial in the accuracy of the whole SLAM system, but modern SLAM systems generally require the front-end to have high real-time and robustness, which puts forward high requirements for the selection and matching of correlation features. Meanwhile, it is also challenging to correctly correlate the sensor data of different modes in time and space because the front end directly manipulates the sensor data. In addition, various degradation scenarios for vision and LiDAR (lack of features, low feature discrimination, and tracking loss caused by fast motion) require the front end to have accurate, reliable, and stable data processing performance.

### 2.3.2 Back-end state estimation

With the idea of minimizing errors, the back-end state estimation optimizes and modifies the initial estimation provided by the front-end globally or locally, so as to obtain more accurate and robust trajectory and three-dimensional environment map. In addition, when the system detects a loop, the back-end module will cooperate with the loop detection module to introduce new constraints to correct the accumulated error, so as to improve the accuracy and robustness of the whole SLAM system. It is necessary to develop more efficient optimization algorithms and data structures to cope with it because the complexity of back-end optimization may increase with the expansion of state and map scales. Meanwhile, nonlinear optimization is easy to fall into local minimum, so it is necessary to set appropriate initial values, optimization strategies and constraints to solve it. In real-time applications, the back-end module needs to complete the optimization process in a limited time, and it is also a challenge how to achieve better optimization results in the shortest time.

### 2.3.3 Loopback detection

Loopback detection is a key component of SLAM, especially in navigation and mapping tasks over long distances or large ranges. However, there is the possibility of misjudgment: one is to identify different scenes as the same scene, and the other is to detect the same scene as different scenes. The main reasons for misjudgment are as follows: (1) The scale inconsistency caused by the change of distance ratio between camera and scene at different time points in visual SLAM. (2) The judgment error caused by the change of viewing angle when observing the same scene at different time points. (3) Dynamic objects may be incorrectly identified as cyclic features, and may also cause changes in the location and appearance of the visited scene. The front-end module of the system may also generate erroneous guidance when tracking dynamic targets. (4) Weather, time, season and other factors may change the characteristics of the same scene. All the above items are all challenges in SLAM loopback detection.

## 2.4 Future development direction of SLAM technology

### 2.4.1 Deep learning-based SLAM

At present, deep learning has shown its potential in the field of SLAM, and there are studies on the introduction and replacement of deep learning methods in each module, including image matching [50, 51], point cloud registration [52], semantic segmentation [53], closed-loop detection [54] and pose estimation [55], etc. In addition, SLAM systems directly based on end-to-end networks [56] also appeared. All the above studies have injected new vitality into the field of SLAM, but so far SLAM methods based on deep learning have not been able to reach the accuracy and reliability of conventional methods. The future development trends of learning-based SLAM systems include: (1) Deep learning networks are needed for online learning on long-term SALM systems in open environments to cope with new scenes and objects independent of training data. (2) Deep learning networks are inseparable from training data. Learning-based SLAM is highly dependent on the richness of training data, requires a lot of labeling

work, and needs to explore low-sample learning techniques. (3) At present, many large models have emerged in the field of deep learning. They have the advantages of powerful data processing capabilities, complex problem solving capabilities, high precision and high performance. Large models are expected to be deployed in SLAM systems to achieve all-round improvement in the future.

#### 2.4.2 Multi-agent collaborative SLAM

Multi-agent refers to the overall system in which various forms of intelligent robots cooperate to complete complex tasks according to task division in a certain time and space [57]. Due to the limitation of the endurance time of a single robot, the efficiency of obtaining 3D information is low with small range; Moreover, it is difficult to comprehensively analyze the complex structure and scene information in real time due to the limitation of working mode. Meanwhile, SLAM has error accumulation characteristics, which makes it difficult to ensure the accuracy of long-term and large-scale mapping. These problems can be solved through the collaborative SLAM of multiple agents. The realization of multi-agent SLAM requires multiple agents to cooperate in a single-machine or cross-machine collaboration manner. Meanwhile, multiple agents share scene maps and perform information interaction and fusion, so as to significantly improve the efficiency, accuracy and robustness of single SLAM.

#### 2.4.3 New type sensors

A variety of new sensors are expected to be introduced into SLAM system with the development of sensor technology. For example, the Event Camera, which is designed to imitate the animal vision system to record the time and location of the event stream. Compared with conventional cameras, it has the advantages of no motion blur, sub-millisecond time delay and ultra-high dynamic range, which has been applied to feature tracking [58], optical flow [59], 3D reconstruction [60], and SLAM [61]. However, due to the uniqueness of event cameras, the processing of noise and spatiotemporal information is different from that of traditional vision, and all task-level algorithms need to be redesigned [62].

# 3 Perception and positioning technology for autonomous driving

## 3.1 The importance of perception and positioning technology in autonomous driving system

In the autonomous driving system, the main task of perception and positioning is to obtain the environmental information around the vehicle through relevant sensors, and determine the position and attitude of the vehicle in the environment, so that the vehicle can achieve safe driving under complex traffic road conditions. Perception technology identifies road conditions, obstacles, traffic signs, and other vehicles based on vehicle sensor data. This kind of understanding of the environment is crucial for the vehicle, because it must be able to dynamically respond to rapid changes on the road, such as avoiding sudden obstacles and adapting to different environmental conditions such as weather and light. Positioning technology can estimate the motion state quantities of the vehicle, including position, pose and speed, in real time and accurately based on the vehicle sensor information, so as to meet the demand of other functional modules of the autonomous driving system for motion state information. Perception and positioning technology provides key underlying information and support for the autonomous driving system, and provides the foundation for the advanced functions of the system such as decision-making and planning, which directly affects the safety, efficiency and reliability of autonomous vehicles. Autonomous driving perception and localization technologies are explained from two aspects: perception and localization. Perception technologies include visual perception, LiDAR perception, and millimeter wave radar sensing, while localization technologies include inertial odometer, satellite navigation and positioning, wheel speed odometer, and map matching. The specific technology is shown in Figure 3.

## 3.2 Autonomous driving perception technology

Real-time, accurate and robust perception of road traffic environment is the basic but most challenging task in autonomous driving. By equipped with multi-modal sensors, the autonomous driving system needs to accurately identify information such as the type, location, trajectory and motion status of targets in road traffic. Autonomous driving perception technology can be mainly divided into visual perception, LiDAR perception and millimeter wave radar perception according to sensor principles.

#### 3.2.1 Visual perception

Vision sensors can obtain images with rich color, texture and semantic information with low cost, so they are widely used in perception tasks of autonomous driving, including traffic target detection, drivable area segmentation and lane line recognition [63]. Object detection ensures the safety of autonomous driving by identifying and locating traffic targets such as vehicles, pedestrians, cyclists, and traffic signs. Object detection methods can be divided into two categories: two-stage networks and single-stage networks. Two-stage networks (such as the R-CNN series, including Fast R-CNN [64] and Faster R-CNN [65]) achieve high accuracy through regional proposal method, but with slower inference. On the other hand, single-stage networks (such as SSD [66] and YOLO [67]) sacrifice partial accuracy in exchange for faster inference speed by simultaneously handling bounding box regression and target classification. Such networks divide input images into meshes or use anchor boxes of various sizes to extract multi-scale features. For autonomous driving scenarios, D. Gragnaniello [68] proposed a 2D multi-object detection and tracking algorithm to solve the problem of multi-class object detection and tracking. OVTrack proposed by Li et al. [69] handles the detection and tracking of arbitrary object classes through visual language models. Huang et al. [70] proposed a multi-object tracking algorithm based on self-supervised appearance model.

Drivable area segmentation enables autonomous vehicles to effectively plan safe trajectories by identifying drivable areas on the road. CNN-based deep learning models perform well in semantic segmentation, which are widely used for pixel-level

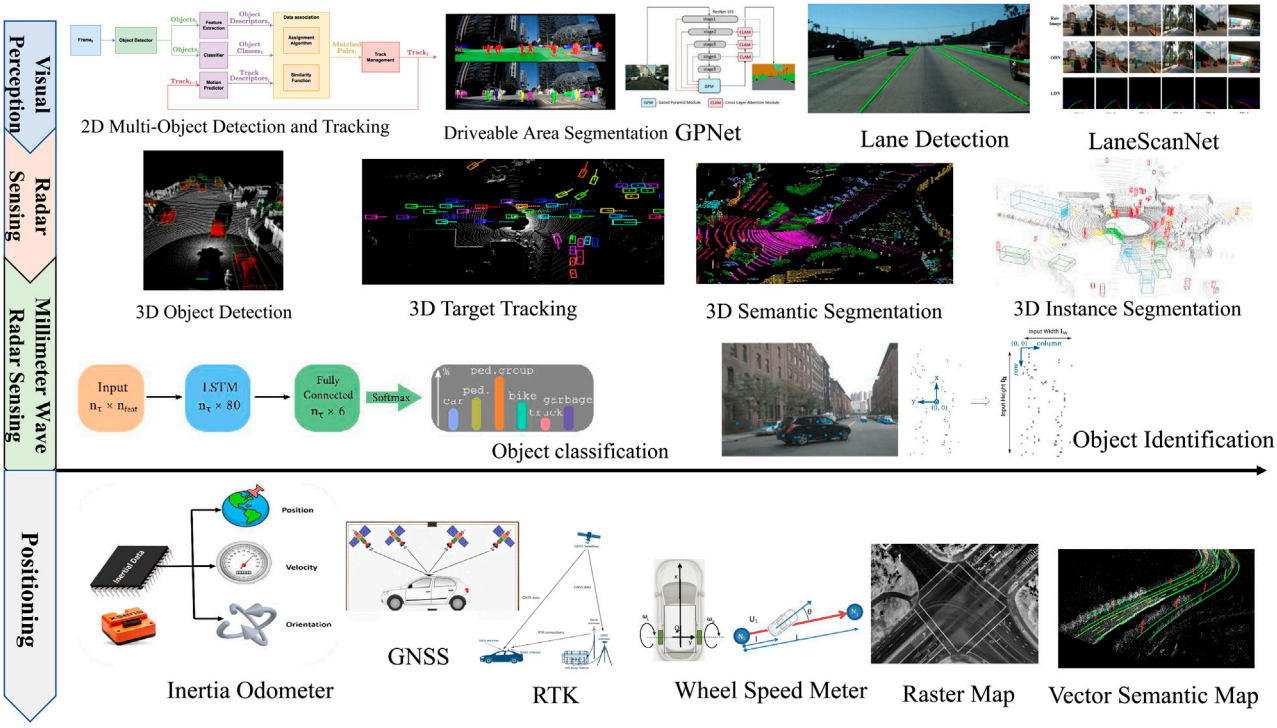



FIGURE 3
Schematic diagram of perception and positioning technology of autonomous driving.

segmentation of drivable regions. Xia and Kim [71] proposed a semantic segmentation architecture that combines multiscale contextual features and low-level features, using hybrid spatial pyramid pooling and global attention fusion. Zhang et al. [72] proposed a GPNet for traffic scene segmentation, combining multi-scale features of gating and pairwise techniques. SegFormer [73] provides sufficient segmentation efficiency and performance through a position-independent hierarchical Transformer encoder and lightweight decoder network. Real-time semantic segmentation is a prerequisite for autonomous driving which needs to achieve competitive segmentation accuracy at low computational costs. DSANet [74] is a computationally efficient network consisting of channel segmentation and shuffling modules and dual attention modules using expanded spatial attention and channel attention to achieve higher segmentation accuracy and lower computational cost.

Accurate lane marker detection and segmentation enable autonomous vehicles to remain within the appropriate lane for precise trajectory control. Conventionally, lane lines are detected using a Canny edge detector [75] and then located in the scene using either a Hough transform [76] or RANSAC [77]. However, these methods are susceptible to illumination and occlusion [78]. CNN-based deep learning models overcome these limitations by annotating lane segments at the pixel level [79]. Zou et al. [80] adopted a segmentation method based on multimodal fusion network for lane detection. Qin et al. [81] proposed an anchor frame-driven sequential classification method for lane detection, which can significantly reduce the computational cost. LaneScanNET [82] assists autonomous driving systems in lane change or lane keeping decisions by combining obstacle detection networks (ODN) and lane detection networks (LDN). The proposed architecture combines the results of obstacle detection and lane line segmentation to predict the obstacle lane state in the field of view of autonomous vehicles. DSUNet [83] is a UNet-based architecture designed for lane detection and path prediction in autonomous driving, using deep separable convolution for faster inference in real-time autonomous driving.

In addition, the visual perception system can obtain different types of information through multiple configurations such as monocular, binocular, and multi-ocular cameras. Such multi-modal data helps to improve the robustness and accuracy of perception, such as obtaining depth information through binocular vision, and achieve panoramic perception through multi-eye vision. However, visual perception technology relies heavily on ambient lighting conditions. The effect of visual perception will be greatly reduced under low light, strong light, backlight and night conditions, requiring additional processing and compensation technology; The visual perception system is easily affected by obstructions, resulting in part of the visual field being blocked. In open environment, visual perception system can provide comprehensive environmental information, but it needs to be used in combination with other sensors in complex environment to make up for the deficiency of visual perception [84].

### 3.2.2 LiDAR perception

LiDAR directly measures the distance of traffic scenes by transmitting and receiving laser beams to obtain high-precision point cloud data. There are different processing methods for point

cloud data. The projection method tries to project the point cloud data into a two-dimensional plane, and then uses a two-dimensional method to process it. Another part of the research voxelizes 3D point cloud data (Voxelization), i.e., the space is divided into small cubes (called voxels). However, a large amount of original information is lost in the process of preprocessing data whether it is the projection method or the voxelization method, and the full performance of high-precision LiDAR cannot be exerted. In order to make full use of the collected information, the mainstream method perception tasks in recent years directly use point cloud data [85]. LiDAR is also used for a variety of perception tasks on autonomous vehicles, such as 3D target detection, 3D target tracking, 3D semantic segmentation, and instance segmentation [86].

LiDAR operates independently of natural light, providing reliable environmental awareness day and night and in various weather conditions. Direct ranging is more accurate and reliable than visual inference of depth information, especially in long-distance and complex scenes. However, the relatively high cost of LiDAR limits its large-scale commercial application, but the cost is expected to decrease with technological progress and expansion of mass production. Due to the large amount of three-dimensional point cloud data generated by LiDAR, it requires powerful data processing capabilities and efficient algorithms for real-time processing and analysis. Therefore, higher requirements are put forward for the computing platform of autonomous driving systems, and data processing pipelines and algorithms need to be optimized to meet real-time needs. And although it can work in a variety of weather conditions, the attenuation and scattering of laser signals may affect the measurement accuracy in extreme environments such as dense fog and heavy rain and snow.

### 3.2.3 Millimeter wave radar sensing

Millimeter-wave radar was mainly used in automotive assisted driving systems in the past. In recent years, with the improvement of semiconductor radio frequency technology, millimeter-wave radar has shown huge advantages in bandwidth, size and cost, and has also shown great application potential in advanced perception tasks of autonomous driving. Scholars have studied the problem of target recognition based on millimeter wave radar point cloud. These methods are divided into two categories: one is to extract information through hand-designed feature extractors, such as Schumann et al. [87] obtain the target area through clustering, and classify pedestrians, vehicles and other targets based on hand-designed multi-dimensional features; The other is to directly extract features through deep neural networks. Danzer et al. used PointNet [88] and PointNet++ [89] methods for pedestrian and vehicle target recognition respectively, and Lombacher et al. [90, 91] converted radar point cloud into rasterized data, and then proposed a series of CNN methods for feature extraction and target recognition.

## 3.3 Autonomous driving positioning technology

Autonomous driving positioning technology can accurately estimate the motion state quantities of the vehicle in real time based on the vehicle sensor information to meet the functional requirements of other autonomous driving modules. The following is an introduction based on the main sensing equipment [92].

### 3.3.1 Inertial odometer

Inertial odometers use the measurement values of inertial devices such as gyroscopes and accelerometers to estimate the carrier's running trajectory. The calculation accuracy depends on the measurement accuracy and stability of inertial devices. For cost considerations, autonomous vehicle platforms usually deploy consumer-grade inertial devices based on Micro Electro Mechanical System (MEMS) structure. MEMS inertial devices often have large measurement noise and complex error characteristics. In order to improve navigation and positioning accuracy, they need to be compensated for their errors in use. The errors of MEMS inertial devices can be roughly divided into static errors, dynamic errors and random errors. Among them, static error and dynamic error are generally considered to be deterministic error related to the motion state of the carrier, and static error can be compensated by offline calibration method [93], or online estimation through other sensor information, while dynamic error is difficult to calibrate or estimate. For random error, it cannot be eliminated by calibration or estimation method, but only an identification model can be established to estimate the parameters of random error. In addition, researchers have explored the application of Gauss-Markov processes [94], wavelet transform methods [95], generalized wavelet moment methods [96] in random error identification and denoising. Due to the complexity of dynamic error and random error models, researchers have begun to try to model inertial odometer errors in a data-driven way in recent years. Martin Brossard et al. [97] employed CNN to predict the bias error of gyroscopes online based on time series window data. Another solution is to directly use neural network to model the calculation process of inertial odometer in an end-to-end way. Joao Paulo et al. [98] encoded the original angular velocity measurement and acceleration measurement input into a discrete CNN channel, and then used a bidirectional Long Short-Term Memory (LSTM) network to encode the time series inertial information to predict the pose increment in an end-to-end manner.

### 3.3.2 Satellite navigation and positioning

The Global Navigation Satellite System (GNSS) uses navigation satellite wireless signals to perform pseudo-range or carrier ranging, calculates the geometric intersection of spatial straight lines based on the ranging information, and estimates the position of the signal receiver in the global coordinate system. GNSS is widely used in location services for autonomous driving due to its simplicity, speed and wide coverage. Standard Point Positioning (SPP), also known as pseudo-range single point positioning, is the most common GNSS positioning method. Influenced by clock error, ionospheric interference, tropospheric interference and other factors, the positioning accuracy of SPP is low. Differential GNSS technology eliminates the temporal and spatial correlation factors such as satellite orbit error, clock error, ionospheric error and tropospheric error by differentiating satellite signals with similar geographical locations, and improves the stability of satellite positioning. In order to improve satellite positioning accuracy, carrier phase ranging technology was born, which can provide centimeter-level ranging accuracy. Combining carrier ranging and

difference principles gave birth to real-time dynamic carrier phase difference technology (Real time Kinematic, RTK) [99], RTK can complete the solution of the ambiguity of the whole circle in a short time and provide position measurement up to centimeter level. At this stage, the RTK technology of reference station network with wide coverage is formed mainly by establishing multiple RTK reference stations for networking and using wireless networks to transmit differential signals.

Satellite signals are easily affected by clock error, clock drift, clock jump, etc. during transmission, which will result in data failure. Therefore, it is necessary to enhance the reliability of self-localization through fault detection. Traffic accidents caused by positioning deviation can be effectively avoided by analyzing the validity of observation data, identifying and eliminating fault data. At present, localization fault detection is usually divided into three categories: snapshot detection, sequence detection, and density anomaly detection [100]. Snapshot detection mode focuses on the consistency test of current observations, which can identify step faults more accurately. The sequence detection method comprehensively uses historical data and current data for consistency test, which can effectively improve the detection effect of slope faults. Furthermore, the distribution uncertainty of observed data and the dependence on prior knowledge can be overcome through identifying anomaly localization data based on the density difference between current data and neighboring data. In the actual operation scenarios of autonomous vehicles, GNSS positioning faces the risk of signal interference; In scenes such as tree-lined road sections, high-rise streets, and under viaducts, blocked by environmental obstacles, the multi-path effect caused by multiple reflections and propagation of satellite signals will greatly interfere with the signal calculation ability of the receiver, resulting in deviations in position and speed measurements. In scenarios such as tunnels and underground garages, satellite signals are completely blocked, and GNSS will completely lose its positioning capabilities [101].

### 3.3.3 Wheel speed odometer

The wheel speed odometer recovers the motion state of the vehicle from the wheel speed information measured by the wheel speed meter. Wheel speed information is essentially the observation information of the vehicle moving speed. Compared with inertial navigation, the number of integrations involved in recovering the vehicle position state through wheel speed is fewer, so the wheel speed odometer is generally more accurate than the inertial odometer. Wheel speed odometers also face the problem of error accumulation, and researchers are also trying to use data-driven methods to improve the accuracy of wheel speed odometers. Uche Onyekpe et al. [102] used the position error between the speed difference model and the GNSS measurement as a neural network supervised signal to train the LSTM network, and the output of the network was used to compensate the position output of the speed difference model; After that, the team further proposed a structurally optimized wheel speed odometry network WhONet [103], using Recurrent Neural Network (RNN) to improve the real-time performance of prediction. Experiments show that the accuracy of this method exceeds the conventional speed differential motion model. Martin Brossard et al. [104] used Gaussian Processes (GP) to model the wheel speed model and its uncertainty, and combined variational inference to train the neural network as the kernel function of GP to reduce the computational complexity of GP.

### 3.3.4 Map matching

Map matching technology matches the positioning features provided by high-precision maps with sensor signals to estimate the position and pose of the vehicle in the map. Different from the SLAM system, high-precision map features are collected through professional mapping equipment, and converted into the global coordinate system through offline optimization and other steps, with excellent position accuracy. Therefore, map matching based on high-precision maps can achieve high-precision global positioning. According to the feature form of map positioning and the type of vehicle sensor, map matching technology has different implementation ideas. In the early autonomous driving, map matching technologies with LiDAR as the main body have been widely studied, such as ICP [29] and NDT [35]. In the grid positioning method proposed by Jesse Levinson et al. [105], the high-precision map records the environmental reflection intensity and elevation information in a plane two-dimensional raster, and the map matching process uses histogram filtering to calculate the likelihood probability corresponding to pose sampling points. Compared with dense point cloud map scheme, vector semantic map models road objects in the environment with parametric geometric vector shapes, and records its geometric attributes and semantic category attributes. Its lightweight characteristics are beneficial to real-time transmission applications of autonomous driving. In addition, compared with conventional visual descriptors, semantic tags, as higher-level abstract information, are less affected by changes in light conditions, seasonal weather changes, and dynamic obstacle occlusion [106]. Therefore, high-precision vector semantic maps have the potential for large-scale application deployment.

## 3.4 Challenges and future development directions of autonomous driving perception and positioning technology

Although the perception and positioning technology of autonomous driving system has made great progress, with the continuous improvement of the intelligence of autonomous driving vehicles, the requirements for corresponding technologies are also constantly increasing, and the current technology still faces some challenges. For example, how does the system maintain the accuracy and robustness of perception in complex scenes and environments such as severe weather like rain and fog, low-recognition scenes with insufficient lighting conditions, and urban congested road sections; Ensure the accuracy and reliability of positioning in GNSS occlusion or denial environments such as tunnels and urban canyons; Strike a balance between real-time performance and computing resource cost when a large number of sensors and computing tasks are involved. In view of these challenging problems, there are the following future development directions.

### 3.4.1 Multi-source fusion sensing and localization

The data of a single sensor will fail in some environments. The current practical solution is to combine multiple complementary sensors to compensate for their respective shortcomings. Different environments rely on different sensor combinations for effective sensing [107]. In the future, multi-sensor fusion will further develop in the direction of multi-modality, scalability and low computing requirements, thereby achieving robust and reliable real-time perception and positioning.

### 3.4.2 Collaborative perception

There are blind spots and limited perception range in the sensor perception of a vehicle. With the continuous development of intelligent network connection technology composed of wireless communication V2X [108] (Vehicle to Everything, including V2V: Vehicle to Infrastructure and V2P: Vehicle to Persons), a new generation of autonomous driving perception technology will further develop to the level of high-dimensional network connection collaborative perception. The information of vehicles, roads, traffic facilities and pedestrians can be shared and interacted through V2X to achieve integrated, global and high-performance traffic status collaborative perception.

### 3.4.3 Unified perspective perception

In recent years, the Bird's Eye View (BEV) [109] unified perception large model based on surround-view camera has attracted a lot of attention from academia and industry, and has become a hot spot in autonomous driving perception research. The BEV perception paradigm converts the information of the vehicle-mounted surround-view sensor into the BEV space through a series of operations, and represents it in the vehicle body coordinate system in the form of a two-dimensional spatial grid. Accordingly, a series of perceptual tasks share the same BEV spatial features, and perform neural network decoding for their respective task objectives. The BEV awareness model is expected to be constructed as a large-parameter neural network model that supports multi-modal, long-time series data input and is oriented to multi-task applications.

# 4 Scene segmentation technology

## 4.1 Application of the definition of scene segmentation technology in 3D data processing

Scene segmentation aims to divide the whole three-dimensional scene into several regions with different semantics, which refers to the category information of real objects observed by scene data. Scene segmentation is the foundation of scene understanding and plays an important role in various fields involving 3D data processing. In autonomous driving, scene segmentation is used to identify roads, vehicles, pedestrians and other obstacles, and generate semantic maps of the surrounding environment of vehicles in real time, providing a basis for decision-making of autonomous driving system; In robotics, scene segmentation helps robots understand their working environment, correctly identify work areas and paths, and enable them to navigate autonomously and interact with the environment; In medical image processing, for three-dimensional CT or MRI data, scene segmentation technology can be used to identify and label different organs and diseased areas, thereby improving the accuracy of diagnosis; In the field of remote sensing mapping, scene segmentation can be used for environmental monitoring, urban modeling and so on.

## 4.2 Classification of scene segmentation techniques

In various applications, most of the objects processed by scene segmentation are represented in the form of point clouds, i.e., the three-dimensional data obtained by scanning and reconstructing the real scene with depth sensors. Since point cloud data is usually disordered, unorganized, and unstructured, and point clouds are huge in open scenarios, it is extremely challenging to segment it and semantically label each point. From the method point of view, semantic segmentation can be divided into: (1) semantic segmentation based on 2D-3D mapping; (2) voxel-based segmentation method; (3) semantic segmentation based on graph convolution; (4) semantic segmentation based on sparse convolution; (5) semantic segmentation based on point convolution. The development route of scene segmentation technology is shown in Figure 4.

### 4.2.1 2D-3D mapping-based method

Compared with three-dimensional computer vision, two-dimensional vision has a longer development history, so in some methods, the semantic segmentation problem of three-dimensional point clouds is tried to be solved by using technologies in the field of two-dimensional vision. V-MVFusion [110] proposes a two-dimensional projection [111] from multiple perspectives to represent a three-dimensional point cloud, and then uses a two-dimensional semantic segmentation network framework [112–114] to process the two-dimensional projection. Based on one-way feature mapping, a bidirectional fusion between two-dimensional features and three-dimensional features is proposed, i.e., two-dimensional image segmentation and three-dimensional point cloud segmentation are performed simultaneously on the scene, and two-way feature mapping is performed in the decoder network, and the experimental results show that bidirectional mapping can improve the performance of semantic segmentation of 3D point clouds better than unidirectional mapping. Because the mapping between point cloud and image often involves preprocessing operations of depth map and occlusion information estimation, the early semantic segmentation methods of point cloud are difficult to be applied in practice. In order to solve this problem, DeepViewAgg [115] proposes a mapping method without preprocessing operations, which can estimate the pixel depth in real time to obtain the correspondence between points and pixels. For the point cloud semantic segmentation method based on 2D-3D mapping, its advantages are that on the one hand, it can make full use of the mature segmentation technology in the field of image, and on the other hand, the 2D image features from multiple perspectives can provide rich context information for 3D semantic segmentation. However, such methods require additional 2D image data and

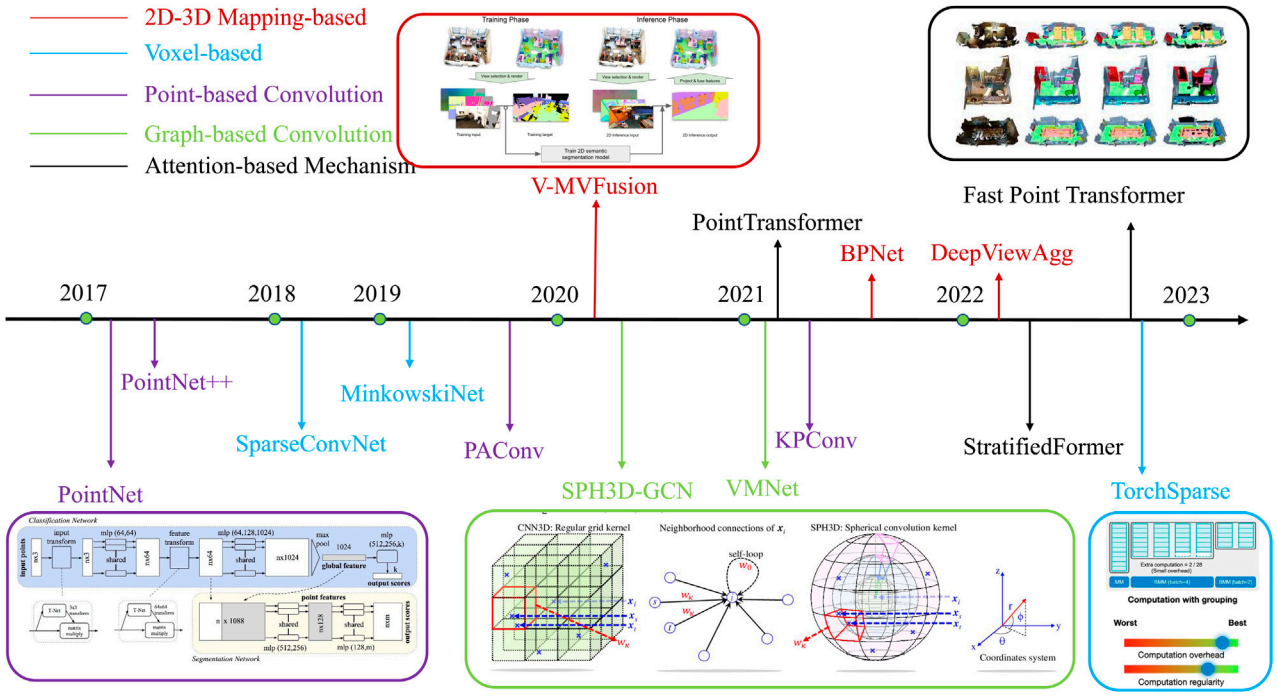



FIGURE 4
Development roadmap of scene segmentation technology.

involve complex multi-view projections, so they rely too much on the choice of camera viewing angle.

### 4.2.2 Voxel-based method

In order to reduce the reliance on redundant image and perspective information, some methods [116, 117] choose to convert point clouds into three-dimensional voxels, i.e., spatially small-volume elements, and then use sparse convolution for semantic segmentation. Sparse convolution concentrates the computation on a non-empty voxel grid, which can effectively reduce the computational overhead. However, since the convolution operation may pass the features of one non-empty voxel to multiple voxels, the number of non-empty voxels will always be high as the multi-layer network is convolved. To solve this problem, SparseConvNet [118] proposes a submanifold sparse convolution operation. This operation requires that for a certain voxel grid point, its non-empty condition is that the central grid point of the receptive field is also non-empty. This method can effectively reduce the number of non-empty voxel grid points, improve the segmentation performance and reduce the computational overhead. After that, more work has been done to try to improve the efficiency and performance of sparse convolution. MinkowskiNet [119] proposes a 4D sparse convolution network, which can process the time series data of three-dimensional point clouds through sparse convolution, and has achieved good results on both indoor and outdoor scene data sets. Haotian et al. [120] further proposed that TorchSparse should be used to improve problems such as computational irregularity and high video memory occupancy in sparse convolution processes. The main advantage of voxel-based method is that it has high efficiency in processing point clouds and is easy to be applied to large-scale point cloud scene data; The disadvantage is that the point cloud needs to be voxelized first. Some key details may be lost when the voxel resolution is low.

### 4.2.3 Point convolution-based method

Also due to the success of convolutional neural networks (CNNs) in the field of images, a lot of work has been done to try to migrate convolutional operations to point clouds. Point cloud segmentation is a technique in computer vision and 3D graphics used to divide point cloud data into different regions or categories. A point cloud is a set of discrete points representing objects or scenes in three-dimensional space, obtained through scanning devices such as LiDAR or 3D cameras. These points typically contain coordinate information (X, Y, Z), and sometimes include additional attributes such as color or intensity. The purpose of point cloud segmentation is to divide these points into meaningful subsets, such as separating buildings, roads, vehicles, and pedestrians from a complex point cloud. PointNet [121] and PointNet++ [122] are representative works in this regard, which aggregate global or local features through max-pooling operations to avoid the negative effects of point cloud disorder. PointNet++ proposes hierarchical local feature aggregation based on PointNet, which is used to improve the network's ability to recognize local features, and lays the foundation for more semantic segmentation methods based on point convolution in the future. KPConv [123] used kernel points to replace the convolution kernel of conventional convolution operations. The features of input points in the convolution process are obtained by the weighted sum of features of adjacent kernel points. Because the kernel points are continuously distributed in the geometric space, their positions can be learned

through the network, and this variable convolution operation can effectively adapt to the problem of uneven local point distribution in the point cloud. On this basis, PAConv [124] used the ScoreNet network to estimate the weighting coefficients of kernel points in the convolution process, and further improved the performance of point convolution through the learnability of the network. The advantage of this method is that it directly processes the point cloud without additional image data or data conversion operations, so it can retain the detailed information of the point cloud to the maximum extent.

### 4.2.4 Graph convolution-based method

Graph convolutional neural network (GNN) is a kind of neural network that specializes in dealing with graph structure. The spatial interaction between three-dimensional points in a point cloud can be represented by a graph, and each point is used as a node of the graph. Therefore, it tries to apply graph convolutional network to point cloud semantic segmentation in some work. L Jiang et al. [125] proposed to enhance point cloud semantic segmentation by an edge feature branch that uses graph convolution techniques to explicitly establish the semantic relationship of each point with its neighborhood points and extract contextual information within the local neighborhood. Similarly, SPH3D-GCN [126] proposes a spherical kernel-based graph convolution operation for point cloud processing, which also directly establishes local graph relations through point coordinates. Another way to use graph convolution to point cloud semantic segmentation is to additionally use the grid model corresponding to the point cloud. Since the grid model has its own coordinate and edge information, graph convolution network can be well applied. DCM-Net [127] proposes to extract geodesic information on the grid model through graph convolution operation, and uses two convolution operations to extract Euclidean distance and geodesic distance respectively, and fuses the two kinds of information through feature stitching. VMNet [128] uses a dual-branch network structure to process point clouds and grid models separately, and proposes a feature fusion module based on attention mechanism to selectively perform fusion, thereby improving the performance of this method in semantic segmentation.

### 4.2.5 Attention mechanism-based method

As attention mechanism shows powerful feature representation capabilities in the fields of natural language processing and computer vision, and it also tried to apply it to semantic segmentation of 3D point clouds in many works. PointTransformer [129] is one of the representative works. Different from previous scalar attention mechanisms, this method proposes a vector attention mechanism for point clouds and uses learnable position coding to improve the network's ability to capture spatial geometric information. However, this method uses a local attention mechanism to reduce the computational overhead. When dealing with complex scenes, it is necessary to superimpose multiple layers of attention modules to expand the receptive field of features. To solve this problem, StratifiedFormer [130] proposes a hierarchical attention mechanism to establish long-distance relationships between features. For each point, this method will simultaneously sample adjacent points in its nearer and farther distances to calculate attention. The sampling is denser in the nearer distance and sparser in the far distance, which can directly expand the receptive field. In addition, some efforts have been made to improve the attention mechanism in point cloud semantic segmentation in terms of efficiency and performance. Fast Point Transformer [131] utilizes a voxel hash architecture to speed up attention modules. Point Transformer V2 [132] groups vector attention on the basis of Point Transformer, further strengthens position coding information, and improves the robustness of network processing point clouds.

## 4.3 Advantages, disadvantages and development trends of existing scene segmentation technologies

Existing scene segmentation technologies are outstanding in high precision and detail capture, which can achieve high-precision segmentation in three-dimensional space, capture subtle geometric details, and provide richer information for the understanding and processing of complex scenes. However, the technology also has some shortcomings. First of all, processing 3D point cloud and voxel data requires a lot of computing resources and high-performance hardware, especially in high-resolution and large-scale scenarios, where computing costs and storage requirements are high. Secondly, it is expensive to obtain high-quality 3D data and perform accurate annotation, and the complexity of data annotation increases the difficulty of preparing training data. Meanwhile, 3D scene segmentation algorithms are usually complex with weak real-time processing capabilities, and are difficult to run efficiently on resource-constrained devices, which is a significant bottleneck in applications that require rapid response. In addition, the existing 3D scene segmentation models lack robustness and generalization ability when they meet complex environments and different scenes, and may require additional tuning and training for specific scenes.

Future development trends mainly focus on the following aspects. First, with the continuous advancement of deep learning technology, especially the application of Transformer and GNN, the accuracy and efficiency of 3D scene segmentation will be further improved. These advanced models are better able to handle large-scale and complex 3D data. Secondly, future research will focus more on multi-task learning and self-supervised learning to reduce the dependence on large-scale labeled data, thereby reducing the cost of data labeling and improving the generalization ability and robustness of the model. Third, with the improvement of hardware performance and the optimization of algorithms, it will be possible to achieve efficient real-time 3D scene segmentation on mobile devices and edge devices, which will promote the practical application of 3D scene segmentation technology in autonomous driving, intelligent robots and other fields. Fourthly, the accuracy and reliability of scene segmentation can be improved by fusing different types of 3D sensor data. Multi-sensor fusion technology will become an important direction of future 3D scene segmentation research. In addition, combining 3D scene information with other modalities (such as text, audio, etc.) can enhance the performance of scene segmentation, and cross-modal fusion technology will provide more comprehensive and accurate information support for 3D scene segmentation.

## 5 Summary and outlook

In this paper, the research status of SLAM, perception and positioning of autonomous driving, scene segmentation and other technologies are introduced respectively. These technologies are interdependent and work together, which constitute the core of modern autonomous driving system. SLAM provides basic positioning and map construction capabilities, scene segmentation provides advanced semantic understanding of the environment, and autonomous driving perception and positioning technology integrates this information for autonomous navigation and decision-making. They are actually closely related and interacted with each other although these technologies belong to different fields on the surface. For example, in the field of autonomous driving, the acquisition of high-precision maps relies on high-precision mapping of SLAM, while higher-level environmental awareness requires scene segmentation and target detection, and real-time positioning also requires SLAM.The future development direction and trend of each technology are prospected when summarizing it. On this basis, development directions applicable to all mentioned technologies will be summarized in the paper, aiming at the common characteristics of all reviewed technologies.

(1) Continuous Application of Deep Learning

Since all the above technologies involve feature extraction and calculation, deep learning has unparalleled advantages in this respect. In the future, deep learning will continue to play a role in various technical fields, integrating more deep learning technologies such as 3D scene reconstruction, 3D target detection, and point cloud completion, which are even expected to completely replace conventional methods in some fields.

(2) Fusion of multi-source and multi-modal information

When the above calculations for three-dimensional data processing or scene perception are applied, multi-source and multi-modal data can provide a more comprehensive and integrated description and understanding of real scenes and objects than single type of data.

(3) High Real-time Performance and Low Computing Load

All data processing technologies will further pursue real-time performance and low computing load to improve processing efficiency and reduce processing costs, so as to promote the real implementation of these technologies in various fields.

The future development of autonomous driving will not only transform transportation technology but will also have profound impacts on law, ethics, and society. In terms of law, the division of responsibility for autonomous vehicle accidents will become a core issue. On the ethical front, the decision-making challenges brought by autonomous driving technology are also a major concern. For example, how should a vehicle make moral judgments when faced with unavoidable accidents (such as the “trolley problem”)? The social impacts are equally important. Autonomous driving could significantly reduce traffic accidents and improve road safety, but it will also disrupt the job market, particularly in the transportation industry. Moreover, as private vehicle ownership declines and shared autonomous vehicle fleets rise, urban planning could be reshaped, changing the way people travel. However, the widespread application of this technology will also raise privacy issues, and how to protect user data will spark ongoing debates in social and policy realms. Overall, the future of autonomous driving is not just about technological breakthroughs but also about comprehensive transformations in law, ethics, and social structures, with the key challenge being how to find a balance in these areas.

## Author contributions

WH: Data curation, Formal Analysis, Investigation, Methodology, Supervision, Writing–review and editing, Conceptualization, Project administration, Writing–original draft. WC: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Supervision, Writing–original draft, Writing–review and editing. ST: Investigation, Validation, Writing–review and editing. LZ: Formal Analysis, Investigation, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded in part by the National Natural Science Foundation of China (No. 62306128), the Basic Science Research Project of Jiangsu Provincial Department of Education (No. 23KJD520003), the Leading Innovation Project of Changzhou Science and Technology Bureau (No. CQ20230072), and the Qingpu District Industry University Research Cooperation Development Foundation of Shanghai (No. 202314).

## Conflict of interest

Author WC was employed by Shanghai Huace Navigation Technology Co., Ltd. Author LZ was employed by Shanghai Future Space-Time Technology Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher’s note

# References


1. Davison AJ, Reid ID, Molton ND, Stasse O. MonoSLAM: real-time single camera SLAM. *IEEE Trans pattern Anal machine intelligence* (2007) 29(6):1052–67. doi:10.1109/tpami.2007.1049

2. Jones ES, Soatto S. Visual-inertial navigation, mapping and localization: a scalable real-time causal approach. *The Int J Robotics Res* (2011) 30(4):407–30. doi:10.1177/0278364910388963

3. Mourikis AI, Roumeliotis SI. A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE (2007). 3565–72.

4. Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. In: *6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE (2007). 225–34.

5. Mur-Artal R, Montiel JMM, Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans robotics* (2015) 31(5):1147–63. doi:10.1109/tro.2015.2463671

6. Rublee E, Rabaud V, Konolige K, Bradski G ORB: an efficient alternative to SIFT or SURF. In: *2011 International conference on computer vision*. Barcelona, Spain: IEEE (2011). 2564–71.

7. Mur-Artal R, Tardós JD. Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans robotics* (2017) 33(5):1255–62. doi:10.1109/tro.2017.2705103

8. Leutenegger S, Lynen S, Bosse M, Siegwart R, Furgale P. Keyframe-based visual–inertial odometry using nonlinear optimization. *The Int J Robotics Res* (2015) 34(3):314–34. doi:10.1177/0278364914554813

9. Campos C, Elvira R, Rodríguez JJG, M. Montiel JM, D. Tardos J. Orb-slam3: an accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans Robotics* (2021) 37(6):1874–90. doi:10.1109/tro.2021.3075644

10. Qin T, Li P, Shen S. Vins-mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Trans robotics* (2018) 34(4):1004–20. doi:10.1109/tro.2018.2853729

11. Forster C, Pizzoli M, Scaramuzza D. SVO: fast semi-direct monocular visual odometry. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE (2014). 15–22.

12. Shen S, Mulgaonkar Y, Michael N, Kumar V Initialization-free monocular visual-inertial state estimation with application to autonomous MAVs. In: *Experimental robotics: the 14th international symposium on experimental robotics*. Cham: Springer International Publishing (2015). p. 211–27.

13. Concha A, Loianno G, Kumar V, Civera J *Visual-inertial direct SLAM 2016 IEEE international conference on robotics and automation (ICRA)*. IEEE (2016). 1331–8.

14. Engel J, Schöps T, Cremers D. *LSD-SLAM: large-scale direct monocular SLAMEuropean conference on computer vision*. Cham: Springer International Publishing (2014). 834–49.

15. Engel J, Koltun V, Cremers D. Direct sparse odometry. *IEEE Trans pattern Anal machine intelligence* (2017) 40(3):611–25. doi:10.1109/tpami.2017.2658577

16. Von Stumberg L, Usenko V, Cremers D. Direct sparse visual-inertial odometry using dynamic marginalization. In: *IEEE international conference on robotics and automation (ICRA)*. IEEE (2018). 2510–7.

17. Bowman SL, Atanasov N, Daniilidis K, Pappas GJ *Probabilistic data association for semantic slam 2017 IEEE international conference on robotics and automation (ICRA)*. IEEE (2017). 1722–9.

18. Lianos KN, Schonberger JL, Pollefeys M, Sattler T Vso: visual semantic odometry. In: *Proceedings of the European conference on computer vision*. Munich, Germany: ECCV (2018). 234–50.

19. Yang S, Scherer S. Monocular object and plane slam in structured environments. *IEEE Robotics Automation Lett* (2019) 4(4):3145–52. doi:10.1109/lra.2019.2924848

20. Frost D, Prisacariu V, Murray D. Recovering stable scale in monocular SLAM using object-supplemented bundle adjustment. *IEEE Trans Robotics* (2018) 34(3):736–47. doi:10.1109/tro.2018.2820722

21. Nicholson L, Milford M, Sünderhauf N. Quadricslam: dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics Automation Lett* (2018) 4(1):1–8. doi:10.1109/lra.2018.2866205

22. Lin S, Wang J, Xu M, Zhao H, Chen Z. Topology aware object-level semantic mapping towards more robust loop closure. *IEEE Robotics Automation Lett* (2021) 6(4):7041–8. doi:10.1109/lra.2021.3097242

23. Julier SJ, Uhlmann JK. New extension of the Kalman filter to nonlinear systemsSignal processing, sensor fusion, and target recognition VI. *Spie* (1997) 3068:182–93. doi:10.1117/12.280797

24. Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans Robotics* (2007) 23(1):34–46. doi:10.1109/tro.2006.889486

25. Godsill S. *Particle filtering: the first 25 years and beyondICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE (2019). 7760–4.

26. Thrun S, Montemerlo M. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The Int J Robotics Res* (2006) 25(5-6):403–29. doi:10.1177/0278364906065387

27. Besl PJ, McKay ND. Method for registration of 3-D shapesSensor fusion IV: control paradigms and data structures. *Spie* (1992) 1611:586–606. doi:10.1117/12.57955

28. Mendes E, Koch P, Lacroix S. ICP-based pose-graph SLAM. In: *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE (2016). 195–200. doi:10.15607/RSS.2014.X.007

29. Zhang J, Singh S. LOAM: lidar odometry and mapping in real-timeRobotics. *Sci Syst* (2014) 2(9):1–9.

30. Shan T, Englot B. *Lego-loam: lightweight and ground-optimized lidar odometry and mapping on variable terrain*. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2018). 4758–65.

31. Deschaud JE. IMLS-SLAM: scan-to-model matching based on 3D data2018. *IEEE Int Conf Robotics Automation (Icra) IEEE* (2018) 2480–5. doi:10.48550/arXiv.1802.08633

32. Behley J, Stachniss C *Efficient surfel-based SLAM using 3D laser range data in urban environmentsRobotics: science and systems*, 2018 (2018). 59.

33. Biber P, Straßer W. The normal distributions transform: a new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2003, 3. IEEE (2003). 2743–8. doi:10.1109/iros.2003.1249285

34. Segal A, Haehnel D, Thrun S. *Generalized-icpRobotics: Sci Syst* (2009) 2(4):435. doi:10.7551/mitpress/8727.003.0022

35. Zhou B, He Y, Qian K, Ma X, Li X. S4-SLAM: a real-time 3D LIDAR SLAM system for ground/watersurface multi-scene outdoor applications. *Autonomous Robots* (2021) 45:77–98. doi:10.1007/s10514-020-09948-3

36. Cohen-OrD ADMNJ. 4-points congruent sets for robust pairwise surface registration. *ACM SIGGRAPH 2008 Pap on-SIGGRAPH* (2008) 8:11–5. doi:10.1145/1399504.1360684

37. Ruan J, Li B, Wang Y, Fang Z *GP-SLAM+: real-time 3D lidar SLAM based on improved regionalized Gaussian process map reconstruction*. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2020). 5171–8.

38. Dube R, Cramariuc A, Dugas D, Sommer H, Dymczyk M, Nieto J, et al. SegMap: segment-based mapping and localization using data-driven descriptors. *The Int J Robotics Res* (2020) 39(2-3):339–55. doi:10.1177/0278364919863090

39. Zhang J, Singh S. Laser–visual–inertial odometry and mapping with high robustness and low drift. *J field robotics* (2018) 35(8):1242–64. doi:10.1002/rob.21809

40. Shan T, Englot B, Ratti C, Rus D Lvi-sam: tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In: *IEEE international conference on robotics and automation (ICRA)*. IEEE (2021). 5692–8.

41. Lin J, Zheng C, Xu W, Zhang F. R LIVE: a robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping. *IEEE Robotics Automation Lett* (2021) 6(4):7469–76. doi:10.1109/lra.2021.3095515

42. Zuo X, Geneva P, Lee W, Liu Y, Huang G Lic-fusion: lidar-inertial-camera odometry. In: *IEEE/RSJ international conference on intelligent robots and systems IROS*. IEEE (2019). 5848–54.

43. Zuo X, Yang Y, Geneva P, Lv J, Liu Y, Huang G, et al. *Lic-fusion 2.0: lidar-inertial-camera odometry with sliding-window plane-feature tracking*. In: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS. IEEE (2020). 5112–9.

44. Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, et al. Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Trans robotics* (2016) 32(6):1309–32. doi:10.1109/tro.2016.2624754

45. Wang W, Wu Y, Jiang Z, Qi J. A clutter-resistant SLAM algorithm for autonomous guided vehicles in dynamic industrial environment. *IEEE Access* (2020) 8:109770–82. doi:10.1109/access.2020.3001756

46. Faessler M, Fontana F, Forster C, Mueggler E, Pizzoli M, Scaramuzza D. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. *J Field Robotics* (2016) 33(4):431–50. doi:10.1002/rob.21581

47. Cheng J, Zhang L, Chen Q, Hu X, Cai J. A review of visual SLAM methods for autonomous driving vehicles. *Eng Appl Artif Intelligence* (2022) 114:104992. doi:10.1016/j.engappai.2022.104992

48. Zou Z. Application of SLAM technology in VR and AR. *AIP Conf Proc AIP Publishing* (2024) 3144(1):030007. doi:10.1063/5.0215525

49. Wang K, Kooistra L, Pan R, Wang W, Valente J. UAV-based simultaneous localization and mapping in outdoor environments: a systematic scoping review. *J Field Robotics* (2024) 41:1617–42. doi:10.1002/rob.22325

50. DeTone D, Malisiewicz T, Rabinovich A. Superpoint: self-supervised interest point detection and description. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2018). 224–36.

51. SAR-optical feature matching: A large-scale patch dataset and a deep local descriptor

52. Wang Y, Solomon JM. Deep closest point: learning representations for point cloud registration. In: *Proceedings of the IEEE/CVF international conference on computer vision* (2019). p. 3523–32.

53. Milioto A, Vizzo I, Behley J, Stachniss C Rangenet++: fast and accurate lidar semantic segmentation. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE (2019). p. 4213–20.

54. Cattaneo D, Vaghi M, Valada A. Lcdnet: deep loop closure detection and point cloud registration for lidar slam. *IEEE Trans Robotics* (2022) 38(4):2074–93. doi:10.1109/tro.2022.3150683

55. Pvn3d: a deep point-wise 3d keypoints voting network for 6dof pose estimation

56. Droid-slam: deep visual slam for monocular, stereo, and rgb-d cameras

57. Lajoie PY, Beltrame G. Swarm-slam: sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics Automation Lett* (2023) 9(1):475–82. doi:10.1109/lra.2023.3333742

58. Kueng B, Mueggler E, Gallego G, Scaramuzza D *Low-latency visual odometry using event-based feature tracks*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2016). 16–23.

59. Benosman R, Clercq C, Lagorce X, Sio-Hoi Ieng Bartolozzi C. Event-based visual flow. *IEEE Trans Neural networks Learn Syst* (2013) 25(2):407–17. doi:10.1109/tnnls.2013.2273537

60. Matsuda N, Cossairt O, Gupta M. Mc3d: motion contrast 3d scanning. In: *2015 IEEE international conference on computational photography (ICCP)*. IEEE (2015). p. 1–10.

61. Zhou Y, Gallego G, Shen S. Event-based stereo visual odometry. *IEEE Trans Robotics* (2021) 37(5):1433–50. doi:10.1109/tro.2021.3062252

62. Gallego G, Delbrück T, Orchard G, Bartolozzi C, Taba B, Censi A, et al. Event-based vision: a survey. *IEEE Trans pattern Anal machine intelligence* (2020) 44(1):154–80. doi:10.1109/tpami.2020.3008413

63. Kerbl B, Kopanas G, Leimkühler T, Drettakis G. 3d Gaussian splatting for real-time radiance field rendering. *ACM Trans Graphics* (2023) 42(4):1–14. doi:10.1145/3592433

64. Girshick R. Fast r-cnn. *Proc IEEE Int Conf Comput Vis* (2015) 1440–8. doi:10.48550/arXiv.1504.08083

65. Ren S, He K, Girshick R, Sun J Faster r-cnn: towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* (2015) 28. doi:10.48550/arXiv.1506.01497

66. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, et al. *Ssd: single shot multibox detector Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, october 11–14, 2016, Proceedings, Part I 14*. Springer International Publishing (2016). p. 21–37.

67. Redmon J, Divvala S, Girshick R, Farhadi A You only look once: unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016). p. 779–88.

68. Gragnaniello D, Greco A, Saggese A, Vento M, Vicinanza A. Benchmarking 2D multi-object detection and tracking algorithms in autonomous vehicle driving scenarios. *Sensors* (2023) 23(8):4024. doi:10.3390/s23084024

69. Li S, Fischer T, Ke L, Ding H, Danelljan M, Yu F Ovtrack: open-vocabulary multiple object tracking. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023). 5567–77.

70. Huang K, Lertniphonphan K, Chen F, Li J, Wang Z Multi-object tracking by self-supervised learning appearance model. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023). 3163–9.

71. Xia Z, Kim J. Mixed spatial pyramid pooling for semantic segmentation. *Appl Soft Comput* (2020) 91:106209. doi:10.1016/j.asoc.2020.106209

72. Zhang Y, Sun X, Dong J, Chen C, Lv Q. GPNet: gated pyramid network for semantic segmentation. *Pattern Recognition* (2021) 115:107940. doi:10.1016/j.patcog.2021.107940

73. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P SegFormer: simple and efficient design for semantic segmentation with transformers. *Adv Neural Inf Process Syst* (2021) 34:12077–90. doi:10.48550/arXiv.2105.15203

74. Elhassan MAM, Huang C, Yang C, Munea TL. DSANet: Dilated spatial attention for real-time semantic segmentation in urban street scenes. *Expert Syst Appl* (2021) 183:115090. doi:10.1016/j.eswa.2021.115090

75. Ding L, Goshtasby A. On the Canny edge detector. *Pattern recognition* (2001) 34(3):721–5. doi:10.1016/s0031-3203(00)00023-6

76. Illingworth J, Kittler J. A survey of the Hough transform. *Computer Vis graphics, image Process* (1988) 44(1):87–116. doi:10.1016/s0734-189x(88)80033-1

77. Choi S, Kim T, Yu W. Performance evaluation of RANSAC family. *J Computer Vis* (1997) 24(3):271–300. doi:10.5244/C.23.81

78. Shyam P, Yoon KJ, Kim KS. Weakly supervised approach for joint object and lane marking detection. *Proc IEEE/CVF Int Conf Comput Vis* (2021) 2885–95. doi:10.1109/ICCVW54120.2021.00323

79. Pan X, Shi J, Luo P, Wang X, Tang X. Spatial as deep: spatial cnn for traffic scene understanding. *Proc AAAI Conf Artif intelligence* (2018) 32(1). doi:10.1609/aaai.v32i1.12301

80. Zou Z, Zhang X, Liu H, Li Z, Hussain A, Li J. A novel multimodal fusion network based on a joint-coding model for lane line segmentation. *Inf Fusion* (2022) 80:167–78. doi:10.1016/j.inffus.2021.10.008

81. Qin Z, Zhang P, Li X. Ultra fast deep lane detection with hybrid anchor driven ordinal classification. *IEEE Trans pattern Anal machine intelligence* (2022) 46(5):2555–68. doi:10.1109/tpami.2022.3182097

82. LaneScanNET: A deep-learning approach for simultaneous detection of obstacle-lane states for autonomous driving systems

83. Lee DH, Liu JL. End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal Image Video Process.* (2023) 17(1):199–205. doi:10.1007/s11760-022-02222-2

84. Wang Y, Du S, Xin Q, He Y, Qian W. Autonomous driving system driven by Artificial intelligence perception fusion. *Acad J Sci Technology* (2024) 9(2):193–8. doi:10.54097/e0b9ak47

85. Zha Y, Shangguan W, Chai L, Chen J. Hierarchical perception Enhancement for different levels of autonomous driving: a review. *IEEE Sensors J* (2024) 24:17366–86. doi:10.1109/jsen.2024.3388503

86. Aung NHH, Sangwongngam P, Jintamethasawat R, Shah S, Wuttisittikulkij L. A review of LiDAR-based 3D object detection via deep learning Approaches towards robust connected and autonomous vehicles. *IEEE Trans Intell Vehicles* (2024) 1–23. doi:10.1109/tiv.2024.3415771

87. Schumann O, Wöhler C, Hahn M, Dickmann J. Comparison of random forest and long short-term memory network performances in classification tasks using radar. In: *2017 sensor data fusion: trends, solutions, applications (SDF)*. IEEE (2017). p. 1–6.

88. Prophet R, Li G, Sturm C, Vossiek M Semantic segmentation on automotive radar maps 2019 IEEE Intelligent Vehicles Symposium (IV). *IEEE* (2019) 756–63. doi:10.1109/IVS.2019.8813808

89. Qi CR, Yi L, Su H, Guibas LJ Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv Neural Inf Process Syst* (2017) 30. doi:10.48550/arXiv.1706.02413

90. Lombacher J, Hahn M, Dickmann J, Wöhler C Object classification in radar using ensemble methods. In: *2017 IEEE MTT-S international conference on Microwaves for intelligent Mobility (ICMIM)*. IEEE (2017). 87–90.

91. Dreher M, Erçelik E, Bänziger T, Knoll A Radar-based 2D car detection using deep neural networks. In: *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)*. IEEE (2020). p. 1–8.

92. Zhao C, Song A, Zhu Y, Jiang S, Liao F, Du Y. Data-driven indoor positioning correction for infrastructure-enabled autonomous driving systems: a lifelong framework. *IEEE Trans Intell Transportation Syst* (2023) 24(4):3908–21. doi:10.1109/tits.2022.3233563

93. Liu A, Tucker R, Jampani V, Makadia A, Snavely N, Kanazawa A. Infinite nature: Perpetual view generation of natural scenes from a single image. In: *Proceedings of the IEEE-CVF International Conference on computer vision*. Montreal, Canada: ICCV (2021).

94. Unsal D, Demirbas K. Estimation of deterministic and stochastic IMU error parameters. In: *Proceedings of the 2012 IEEE/ION position, location and navigation symposium*. IEEE (2012). 862–8.

95. Dong P, Cheng J, Liu L, Zhang W *Application of improved wavelet de-noising method in MEMS-IMU signals 2019 Chinese Control Conference (CCC)*. IEEE (2019). 3881–4.

96. Radi A, Sheta B, Nassar S, Arafa I, Youssef A, El-Sheimy N Accurate identification and implementation of complicated stochastic error models for low-cost MEMS inertial sensors. In: *2020 12th international conference on Electrical Engineering (ICEENG)*. IEEE (2020). 471–5.

97. Brossard M, Bonnabel S, Barrau A. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics Automation Lett* (2020) 5(3):4796–803. doi:10.48550/arXiv.2002.10718

98. Silva do Monte Lima JP, Uchiyama H, Taniguchi R. End-to-end learning framework for imu-based 6-dof odometry. *Sensors* (2019) 19(17):3777. doi:10.3390/s19173777

99. Ku J, Mozifian M, Lee J, Harakeh A, Waslander SL *Joint 3d proposal generation and object detection from view aggregation 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (2018). p. 1–8.

100. Wang W, Shangguan W, Liu J, Chen J. Enhanced fault detection for GNSS/INS integration using maximum correntropy filter and local outlier factor. *IEEE Trans Intell Vehicles* (2023) 9:2077–93. doi:10.1109/tiv.2023.3312654

101. Hou P, Zha J, Liu T, Zhang B. Recent advances and perspectives in GNSS PPP-RTK. *Meas Sci Technology* (2023) 34(5):051002. doi:10.1088/1361-6501/acb78c

102. Onyekpe U, Palade V, Kanarachos S, Christopoulos SRG *Learning uncertainties in wheel odometry for vehicular localisation in GNSS deprived environments*. In: 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE (2020). 741–6.

103. Onyekpe U, Palade V, Herath A, Kanarachos S, Fitzpatrick ME. WhONet: wheel Odometry neural Network for vehicular localisation in GNSS-deprived environments. *Eng Appl Artif Intelligence* (2021) 105:104421. doi:10.1016/j.engappai.2021.104421

104. Brossard M, Bonnabel S. *Learning wheel odometry and IMU errors for localization2019 international conference on robotics and automation (ICRA)*. IEEE (2019). 291–7.

105. Levinson J, Thrun S. *Robust vehicle localization in urban environments using probabilistic maps2010 IEEE international conference on robotics and automation*. IEEE (2010). p. 4372–8.

106. Xiao Z, Yang D, Wen T, Jiang K, Yan R. Monocular localization with vector HD map (MLVHM): a low-cost method for commercial IVs. *Sensors* (2020) 20(7):1870. doi:10.3390/s20071870

107. Ye X, Song F, Zhang Z, Zeng Q. A review of small UAV navigation system based on multi-source sensor fusion. *IEEE Sensors J* (2023) 23:18926–48. reinforcement learning. doi:10.1109/jsen.2023.3292427

108. Chen S, Hu J, Shi Y, Peng Y, Fang J, Zhao R, et al. Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G. *IEEE Commun Stand Mag* (2017) 1(2):70–6. doi:10.1109/mcomstd.2017.1700015

109. Ma Y, Wang T, Bai X, Yang H, Hou Y, Wang Y, et al. Vision-centric bev perception: a survey. *IEEE Trans Pattern Anal Mach Intell* (2024). doi:10.1109/TPAMI.2024.3449912

110. Kundu A, Yin X, Fathi A, Ross D, Brewington B, Funkhouser T, et al. Virtual multi-view fusion for 3d semantic segmentation. In: *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*. Springer International Publishing (2020). 518–35.

111. Lawin FJ, Danelljan M, Tosteberg P, Bhat G, Khan FS, Felsberg M Deep projective 3D semantic segmentation/computer analysis of images and Patterns. In: *17th international conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I 17*. Springer International Publishing (2017). 95–107.

112. Huang J, Zhang H, Yi L, Funkhouser T, Nießner M, Guibas L Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019). 4440–9.

113. Tatarchenko M, Park J, Koltun V, Zhou QY Tangent convolutions for dense prediction in 3d. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018). 3887–96.

114. Hu W, Zhao H, Jiang L, Jia J, Wong TT Bidirectional projection network for cross dimension scene understanding. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021). 14373–82.

115. Robert D, Vallet B, Landrieu L. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022). p. 5575–84.

116. Graham B. Sparse 3D convolutional neural networks. (2015) 150.1–9. doi:10.5244/c.29.150

117. Engelcke M, Rao D, Wang DZ, Tong CH, Posner I Vote3deep: fast object detection in 3d point clouds using efficient convolutional neural networks. In: *IEEE international conference on robotics and automation (ICRA)*. IEEE (2017). 1355–61.

118. Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018). 9224–32.

119. Choy C, Gwak JY, Savarese S. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019). 3075–84.

120. Tang H, Liu Z, Li X, Lin Y, Han S Torchsparse: efficient point cloud inference engine. *Proc Machine Learn Syst* (2022) 4:302–15. doi:10.48550/arXiv.2204.10319

121. Qi CR, Su H, Mo K, Guibas LJ Pointnet: deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017). 652–60.

122. Qi CR, Yi L, Su H, Guibas LJ Pointnet+: deep hierarchical feature learning on point sets in a metric space. *Adv Neural Inf Process Syst* (2017) 30.

123. Thomas H, Qi CR, Deschaud JE, Marcotegui B, Goulette F, Guibas LJ Kpconv: Flexible and deformable convolution for point clouds. In: *Proceedings of the IEEE/CVF international conference on computer vision* (2019). 6411–20.

124. Xu M, Ding R, Zhao H, Qi X Paconv: position adaptive convolution with dynamic kernel assembling on point clouds. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021). 3173–82.

125. Jiang L, Zhao H, Liu S, Shen X, Fu CW, Jia J Hierarchical point-edge interaction network for point cloud semantic segmentation. *Proc IEEE/CVF Int Conf Computer Vis* (2019) 10433–41. doi:10.1109/ICCV.2019.01053

126. Lei H, Akhtar N, Mian A. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE Trans pattern Anal machine intelligence* (2020) 43(10):3664–80. doi:10.1109/tpami.2020.2983410

127. Schult J, Engelmann F, Kontogianni T, Leibe B *Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020). 8612–22.

128. Hu Z, Bai X, Shang J, Zhang R, Dong J, Wang X, et al. Vmnet: voxel-mesh network for geodesic-aware 3d semantic segmentation. *Proc IEEE/CVF Int Conf Computer Vis* (2021) 15488–98. doi:10.48550/arXiv.2107.13824

129. Zhao H, Jiang L, Jia J, Torr P, Koltun V Point transformer. In: *Proceedings of the IEEE/CVF international conference on computer vision* (2021). 16259–68.

130. Lai X, Liu J, Jiang L, Wang L, Zhao H, Liu S, et al. Stratified transformer for 3d point cloud segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022). 8500–9.

131. Park C, Jeong Y, Cho M, Park J Fast point transformer. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022). 16949–58.

132. Wu X, Lao Y, Jiang L, Liu X, Zhao H Point transformer v2: Grouped vector attention and partition-based pooling. *Adv Neural Inf Process Syst* (2022) 35:33330–42. doi:10.48550/arXiv.2210.05666

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 21 October 2024
DOI 10.3389/fphy.2024.1439122



OPEN ACCESS

EDITED BY
Haoyu Chen,
University of Oulu, Finland

REVIEWED BY
Yang Yang,
Yunnan Normal University, China
Tanmoy Chakraborty,
Sharda University, India
Zhiqin Zhu,
Chongqing University of Posts and Telecommunications, China
Shireen Y. Elhabian,
The University of Utah, United States

*CORRESPONDENCE
Fan Li,
478263823@qq.com

RECEIVED 27 May 2024
ACCEPTED 07 October 2024
PUBLISHED 21 October 2024

CITATION
Zhao W, Li F, Diao Y, Fan P and Chen Z (2024) Cap2Seg: leveraging caption generation for enhanced segmentation of COVID-19 medical images.
*Front. Phys.* 12:1439122.
doi: 10.3389/fphy.2024.1439122



# Cap2Seg: leveraging caption generation for enhanced segmentation of COVID-19 medical images

Wanlong Zhao[1,2], Fan Li[1,2]*, Yueqin Diao[1,2], Puyin Fan[1,2] and Zhu Chen[1,2]

[1]Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China, [2]Yunnan Key Laboratory of Artificial Intelligence, Kunming, China

Incorporating medical text annotations compensates for the quality deficiencies of image data, effectively overcoming the limitations of medical image segmentation. Many existing approaches achieve high-quality segmentation results by integrating text into the image modality. However, these approaches require matched image-text pairs during inference to maintain their performance, and the absence of corresponding text annotations results in degraded model performance. Additionally, these methods often assume that the input text annotations are ideal, overlooking the impact of poor-quality text on model performance in practical scenarios. To address these issues, we propose a novel generative medical image segmentation model, Cap2Seg (Leveraging Caption Generation for Enhanced Segmentation of COVID-19 Medical Images). Cap2Seg not only segments lesion areas but also generates related medical text descriptions, guiding the segmentation process. This design enables the model to perform optimal segmentation without requiring text input during inference. To mitigate the impact of inaccurate text on model performance, we consider the consistency between generated textual features and visual features and introduce the Scale-aware Textual Attention Module (SATaM), which reduces the model's dependency on irrelevant or misleading text information. Subsequently, we design a word-pixel fusion decoding mechanism that effectively integrates textual features into visual features, ensuring that the text information effectively supplements and enhances the image segmentation task. Extensive experiments on two public datasets, MosMedData+ and QaTa-COV19, demonstrate that our method outperforms the current state-of-the-art models under the same conditions. Additionally, ablation studies have been conducted to demonstrate the effectiveness of each proposed module. The code is available at https://github.com/AllenZzzzzzzz/Cap2Seg.

KEYWORDS

**COVID-19, vision-language, multi-task learning, medical image segmentation, medical image captioning**

# 1 Introduction

COVID-19 has rapidly become a global epidemic since the early 2020s Benvenuto et al. [1]. Within 6 months of the outbreak, over 1.5 million cases of COVID-19 had been reported worldwide, with more than 92,000 deaths Organization et al. [2]. Clinically, reverse transcription polymerase chain reaction (RT-PCR) is the standard method for diagnosing COVID-19. Still, it has drawbacks, such as a high false-negative rate Chan et al. [3] and an inability to provide information about the patient's condition. Computed tomography (CT), due to its convenience and ability to display the three-dimensional structure of the lungs, has been considered an essential complement to RT-PCR testing for the early diagnosis of COVID-19, especially in the follow-up assessment and evaluation of disease progression Raoof and Volpi [4]. Consequently, the automatic segmentation of lung infections in CT scans using computer vision techniques has garnered widespread attention from clinical researchers Shi et al. [5].

With the advent of deep learning, medical image segmentation has become a hot topic in computer vision researchZhu et al. [6]. This task focuses on identifying pixel features of anatomical or pathological regions from the background of medical images and applying these features to the image segmentation process Liu et al. [7]; Zhu et al. [8]. Consequently, many deep learning systems have been proposed for COVID-19 infection detection Ronneberger et al. [9]; Zhou et al. [10], achieving state-of-the-art performance Wang et al. [11]; Fan et al. [12]. Figure 1A illustrates that the encoder-decoder architecture is a more commonly used approach. In this architecture, the encoder is responsible for extracting image features, while the decoder restores these features to the original image size and produces the final segmentation results.

However, the aforementioned traditional pixel-wise supervised automatic segmentation methods based on deep learning neglect the semantic information in medical reports. Medical reports often contain information about the lesion areas, such as size and quantity, which can complement image data and provide additional supervisory signals for diagnosis Monajatipoor et al. [13]. Vision-language models have been extensively researched recently and achieved remarkable results in cross-modal tasks. Consequently, many studies have begun exploring combining textual information from medical reports with the segmentation process to improve segmentation accuracy Li et al. [14]; Chen et al. [15]; Huemann et al. [16]; Tomar et al. [17]. As shown in Figure 1B, a typical multimodal medical image segmentation research workflow first relies on two specially designed encoders to extract visual and language features separately. These extracted features are then integrated using a specific fusion strategy and processed through a network decoder intended explicitly for multimodality to obtain the segmentation results.

Although vision-language models have shown promising performance in the segmentation field, they face two significant challenges in practical applications within the medical domain. Firstly, these methods Li et al. [14]; Huemann et al. [16]; Wen et al. [18], trained using image-text pairs, often experience performance degradation during inference if the text is unavailable. This creates a dependency on image-text pairs. In real-world scenarios, this form of inference frequently contradicts the process of the model independently assisting clinical diagnosis: it

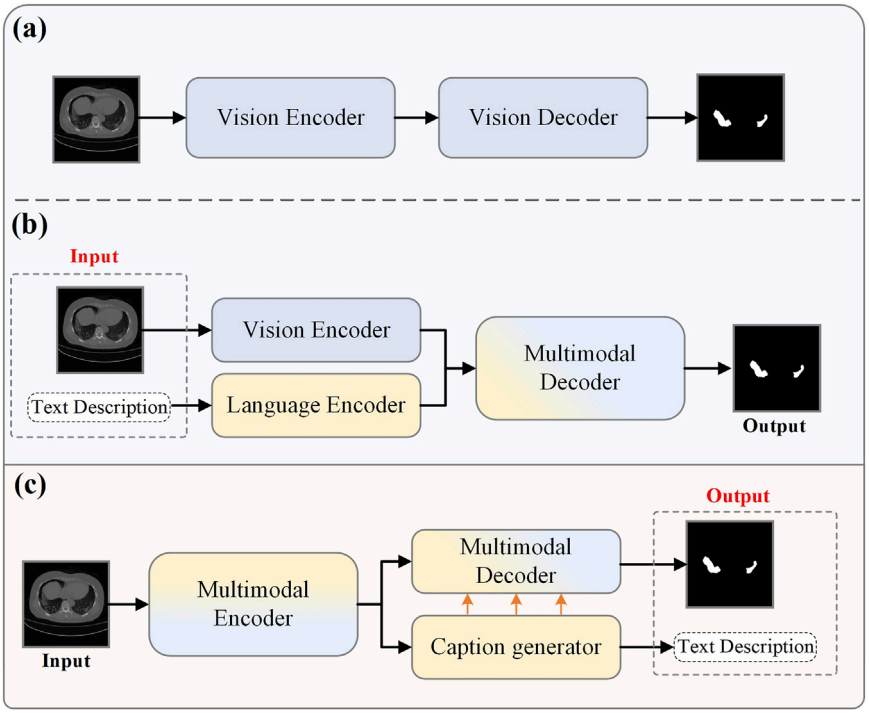



FIGURE 1
Current medical image segmentation models. **(A)** Traditional medical image segmentation. **(B)** Vision-Language multimodal medical image segmentation. **(C)** Our proposed model in this paper.

is usually challenging to obtain textual information from medical reports before the doctor completes the diagnosis Li et al. [19]; Yu et al. [20]. This means that if the model relies on these finalized reports to enhance its performance, it is essentially duplicating the diagnosis already made by the doctor rather than providing an independent auxiliary diagnosis. This dependency significantly diminishes the model's auxiliary value and deviates from its original purpose of independently aiding medical diagnosis. Secondly, existing vision-language models Wen et al. [21] often focus solely on effectively combining text and visual modalities, neglecting text accuracy's impact on model performance. Inaccurate text can mislead the model and negatively affect its performance. In practical applications, medical reports may contain errors due to various factors. Effectively handling this imperfect textual information and preventing it from impairing model performance is also a significant challenge.

In summary, there are two main challenges: 1. How to address the model's dependency on image-text pairs during the inference stage; 2. How to mitigate the impact of text accuracy on model performance. To solve the first challenge, we propose the Cap2Seg model, as shown in Figure 1C. This model combines the image captioning task and requires only a lesion image as input to simultaneously output segmentation results and corresponding text descriptions, successfully eliminating the model's dependency on image-text pair data. Considering that some generated texts may sometimes deviate from actual medical reports and potentially affect segmentation performance, we designed a Scale-aware Textual Attention Module (SATaM) and a semantic consistency loss (SCloss) function to address the second challenge. These two mechanisms work together to ensure that the attention of the generated language features is focused on the lesion areas, effectively avoiding misleading the model with biased generated texts. Additionally, we introduced a Language-Aware Visual Decoder (LAVD), which effectively integrates multi-scale language features with visual features and decodes them, significantly improving the overall quality of the segmentation results. Our contributions are summarized as follows.

(1) The proposed Cap2Seg combines caption generation with lesion area segmentation, generating related medical text descriptions simultaneously. Leveraging the generated textual information to supplement the segmentation task effectively improves the accuracy of medical image segmentation. This eliminates the model's dependency on image-text pairs and provides additional references for clinical diagnosis.
(2) The SATaM optimizes the quality of language features and enhances the model's ability to handle textual biases, thereby improving overall robustness. Concurrently, the Language-Aware Visual Decoder (LAVD) effectively integrates visual and linguistic features, significantly improving segmentation quality.
(3) Experiments conducted on two publicly available COVID-19 datasets demonstrate that our proposed method outperforms most state-of-the-art models in segmentation performance.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review and summary of previous research related to our work. Section 3 describes the architecture of the proposed network. In Section 4, we present and analyze the experimental results. Finally, in Section 5, we conclude our work.

# 2 Related works

This section reviews and summarizes previous relevant studies related to our work, focusing on Visual-Language image segmentation, image captioning, and multi-task learning.

## 2.1 Visual-language image segmentation

In recent years, multimodal segmentation techniques that combine visual and language modalities have garnered extensive attention. Hu et al. Hu et al. [22] pioneered using textual descriptions to assist image segmentation, sparking further research into effectively integrating visual and textual information to enhance segmentation results. Broadly, this task can be categorized into two types: referring image segmentation in natural scenes and image segmentation in medical contexts.

### 2.1.1 Referring image segmentation

In applications within natural settings, early studies Liu et al. [23]; Li et al. [24]; Shi et al. [25]; Ye et al. [26] focused on developing more effective techniques for extracting and merging visual and linguistic features. Liu et al. Liu et al. [23] introduced a multimodal Long Short-Term Memory network specifically designed to process and fuse multimodal features of each word. Shi et al. Shi et al. [27] proposed a keyword-aware network that, while extracting text features, assigns higher weights to keywords, thereby improving the model's ability to recognize text-indicated objects. The introduction of attention mechanisms paved new ways for effective cross-modal feature fusion. Ye et al. Ye et al. [26] employed non-local blocks Wang et al. [28] to design a cross-modal self-attention module for integrating features across modalities. Similarly, other studies Chen et al. [29]; Hu et al. [30]; Shi et al. [27]; Chen et al. [31] utilized various attention mechanisms to process and integrate cross-modal features. Unlike these later fusion approaches, LAVT Yang et al. [32] achieved an early fusion of linguistic and visual features at the intermediate layers of a Transformer network, enhancing cross-modal alignment and the model's integration of visual and linguistic information. With the significant rise of CLIP Radford et al. [33] in the multimodal field, some research began to explore using contrastive learning to represent cross-modal data, such as LSeg Li et al. [34] and GroupViT Xu et al. [35]. These studies leveraged the advanced representational capabilities of CLIP in multimodal scenarios, effectively enhancing image segmentation efficiency and accuracy and demonstrating exceptional capabilities in zero-shot inference scenarios. Further research has focused on the role of text structure in enhancing multimodal information processing. Yu et al. Yu et al. [36] and Huang et al. Huang et al. [37] utilized sentence structure knowledge to capture concepts within multimodal features, such as categories, attributes, and relationships. Hui et al. Hui et al. [38] used syntactic structures between words to guide multimodal context aggregation. Ding et al.

Ding et al. [39] introduced a dynamic query generation module capable of dynamically producing multiple queries based on the input text to accommodate diverse linguistic scenarios, making multimodal information fusion more targeted and specific.

### 2.1.2 Medical image segmentation

In the medical field, Li et al. Li et al. [14] proposed the LViT model, a hybrid of CNNs and Transformers, which incrementally integrates medical text annotations into the image segmentation process to compensate for the quality deficiencies of image data. Unlike LViT, Bi-VLGM Chen et al. [15] emphasizes maintaining consistency within modal features and uses a visual-language graph matching module to handle the category-severity relationships between visual and text features, enabling the segmentation model to learn valuable representations selectively. Other studies Huang et al. [40]; Zhang et al. [41]; Huemann et al. [16]; Dai et al. [42] have used more flexible medical reports for segmentation. ConTEXTualNet Huemann et al. [16] employs attention mechanisms to decode image features based on text in medical reports, guiding the model to focus on text-related image pixels. Some methods Tomar et al. [17], even without available medical reports or texts, utilize auxiliary classification tasks to embed textual attributes (size and number) during encoding. This approach enables the network to adapt to various sizes and numbers of polyp cases, thereby enhancing segmentation performance. However, existing state-of-the-art methods Li et al. [14]; Chen et al. [15] still rely on matched medical text and image data during the inference stage to achieve optimal performance. Their performance may suffer when only image input is available without corresponding text. In contrast, the Cap2Seg model proposed in this study requires only one image to achieve optimal performance during inference.

## 2.2 Image captioning

Image captioning, which aims to produce natural language descriptions based on static visual content Vinyals et al. [43]; Ghandi et al. [44], represents a challenging cross-modal translation task Zhang et al. [45]; Yu et al. [46]. This task demonstrates particular application value in the medical field Li et al. [47]; Hou et al. [48]; Wang et al. [49]. For instance, Li et al. Li et al. [47] developed a Knowledge-driven Encoding, Retrieval, and Paraphrasing (KERP) model to improve medical image descriptions. Our research focuses not on designing a new captioning model *per se* but on employing image caption generation as an auxiliary module. To the best of our knowledge, this study is the first attempt to explore caption generation in medical image segmentation.

## 2.3 Multi-task learning

Multi-task learning aims to enhance the performance of individual or multiple tasks by jointly training related tasks, utilizing the correlations and shared information between them for mutual benefit. For example, Wu et al. Wu et al. [50] introduced the CGG framework, which combines image caption generation and referring image segmentation tasks. This framework employs caption generation loss to supervise the model, improving image segmentation quality. Similarly, Sun et al.'s PFOS model Sun et al. [51], which integrates the tasks of Referring Expression Comprehension and Generation, leverages cross-attention and multimodal fusion mechanisms to boost overall model performance significantly. Moreover, Zhang et al. Zhang et al. [52] demonstrated significant performance improvements in medical image analysis by combining gastric cancer segmentation with lymph node classification tasks, effectively managing the inter-task relationships and heterogeneity through multi-scale features and refined attention mechanisms. Following this concept, Cap2Seg merges the functions of image caption generation and medical image segmentation. The goal is to utilize the generated textual annotations as supplementary information to the image modality, thereby enhancing the performance of the segmentation task.

# 3 Proposed method

This section elaborates on the proposed method, encompassing four components: the Multimodal Synergistic Dual-Flow Encoder (MSDFE) module, the Multimodal Semantic Enhancement and Captioning Module (MSECM), the Scale-aware Textual Attention Module (SATaM), and the Language-Aware Visual Decoder (LAVD).

## 3.1 Overview

The caption-driven multimodal COVID-19 segmentation framework proposed in this paper is illustrated in Figure 2A. This framework addresses two primary tasks: medical image captioning and medical image lesion segmentation. The MSDFE module initially processes the input image, mapping it into a multimodal space that combines visual and textual data, thereby providing a comprehensive set of features for both tasks. The MSECM then refines these features to enhance their relevance to each task. Concurrently, the SATaM and Semantic Consistency Loss (SCloss) are employed to apply focused attention to the textual features, thereby minimizing the model's reliance on non-relevant or potentially misleading information and concentrating efforts on lesion areas. Finally, the LAVD integrates and upsamples the textual and visual features to produce the final segmentation results. In summary, our proposed framework leverages the synergistic effects of multitask learning to exploit the rich complementary information contained in generated text annotations, thereby enhancing the segmentation quality of COVID-19 and providing additional textual diagnostic support.

## 3.2 Multimodal synergistic dual-flow encoder

Given the high variability in the shape, size, and location of COVID-19 infection-related issues, and the requirement for the Cap2Seg model to perform both image segmentation and image captioning tasks, extracting richer features from the input images is crucial. Convolutional Neural Networks (CNN) can accumulate

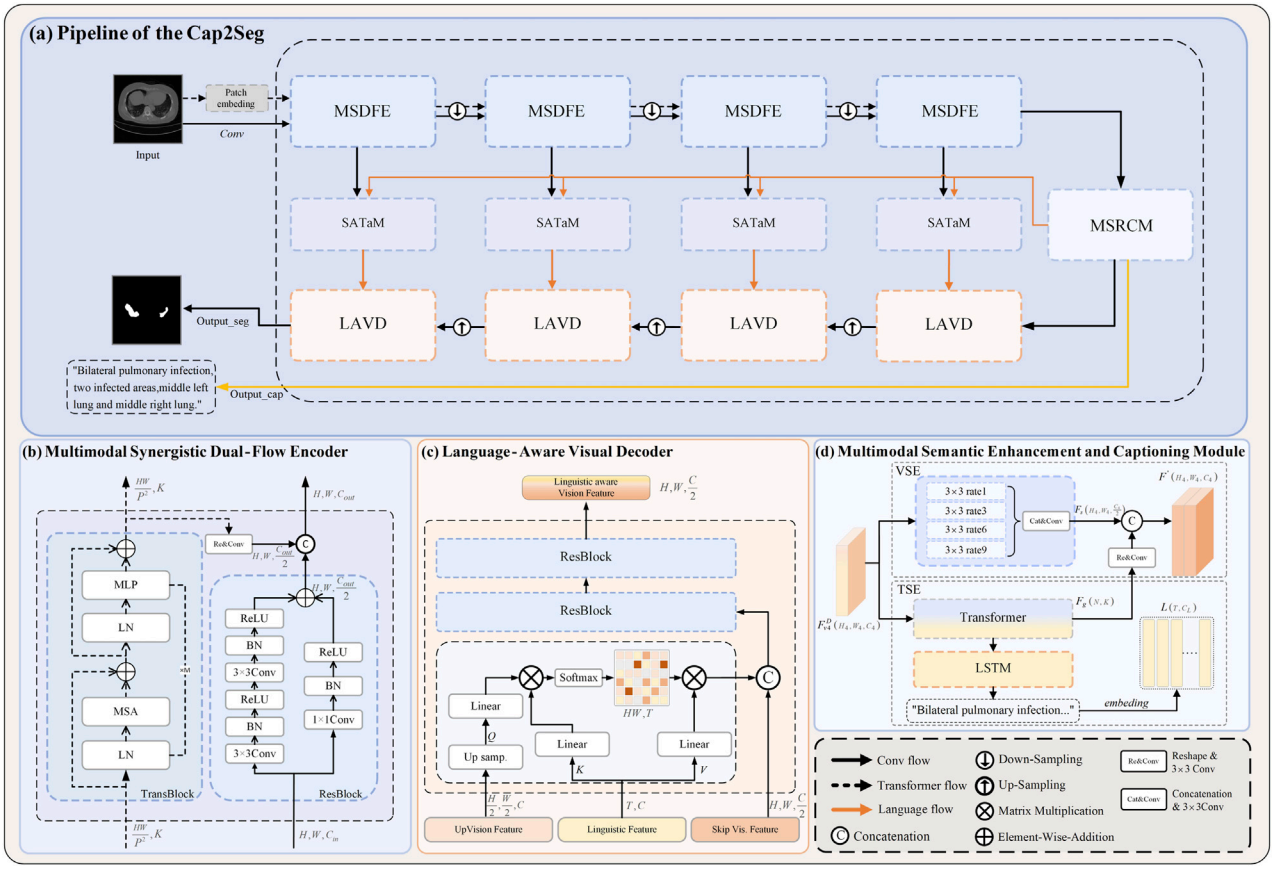



FIGURE 2
Overall framework of the proposed method. **(A)** The leading network comprises the following components: **(B)** The multimodal synergistic Dual-Flow Encoder (MSDFE) module, **(C)** the Multimodal Semantic Enhancement and Captioning Module (MSECM), the Scale-aware Textual Attention Module (SATaM), and **(D)** the Language-Aware Visual Decoder (LAVD).

spatial information of images, focusing on capturing local information such as the texture and contours of lesion areas. At the same time, the self-attention mechanism can explore long-range dependencies in images, focusing on capturing global information. To fully extract diverse features, this paper proposes a Multimodal Synergistic Dual-Flow Encoder (MSDFE), which combines the strengths of CNN and Transformer. As shown in Figure 2B, MSDFE consists of two parallel feature extraction branches: the first branch is the "trans flow" processed by TransBlock (indicated by dashed lines in the figure), and the second branch is the "conv flow" processed by ResBlock (indicated by solid lines in the figure). MSDFE can extract local, global, and long-range dependency features from images through this combination, thus providing a more expressive feature set for both tasks.

Specifically, The ResBlock comprises a pair of $3 \times 3$ convolutional blocks, each succeeded by batch normalization Ioffe and Szegedy [53] and the ReLU activation function Nair and Hinton [54]. The architecture is finalized with a residual connection featuring $1 \times 1$ convolution that synergistically integrates the input with the convolutional layers' outputs, as specified in the following Equations 1, 2:

$$\tilde{x}_{\text{out}} = \sigma\left(BN\left(\text{Conv}_{3\times3}\left(x_{\text{in}}\right)\right)\right) \tag{1}$$

$$x_{\text{out}} = \sigma\left(BN\left(\text{Conv}_{3\times3}\left(\tilde{x}_{\text{out}}\right)\right)\right) + \sigma\left(BN\left(\text{Conv}_{1\times1}\left(x_{\text{in}}\right)\right)\right) \tag{2}$$

In this context, $\sigma$ denotes the ReLU activation function, $BN$ stands for Batch Normalization, $\text{Conv}_{3\times3}$ and $\text{Conv}_{1\times1}$ are the convolutions of size $3 \times 3$ and $1 \times 1$, respectively.

As for TransBlock, it initially processes the input image $x \in \mathbb{R}^{(H,W,C)}$ into flattened uniform non-overlapping patches $x_p \in \mathbb{R}^{(P^2\times C,N)}$, where $(H, W, C)$ are the input image's resolution and channels, $(P, P)$ is the resolution per image patch, and $N = HW/P^2$ is the number of patches. These patches are then mapped onto a $k$-dimensional embedding space $z_0$ by a trainable linear layer $E \in \mathbb{R}^{(P^2\times C,K)}$. The definition of $z_0$ is provided as follows in Equation 3:

$$z_0 = \left[x_p^1 E; x_p^2 E; \ldots; x_p^N E\right] \tag{3}$$

Subsequently, the embedded feature $z_0 \in \mathbb{R}^{(N,K)}$ serves as the input for TransBlock, comprising a Multi-Head Self Attention module followed by a 2-layer MLP with interposed with a GELU activation function. A LayerNorm layer is applied before each MAS module and each MLP, and a residual connection is applied after each module Dosovitskiy et al. [55]; Vaswani et al. [56]. Which can be expressed as Equations 4, 5:

$$z_i' = MSA\left(LN\left(z_{i-1}\right)\right) + z_{i-1}, \quad i = 1 \ldots M \tag{4}$$

$$z_i = MLP\left(LN\left(z_i'\right)\right) + z_i', \quad i = 1 \ldots M \tag{5}$$

Herein, $MSA$ signifies multi-head self-attention, $LN(\cdot)$ denotes layer normalization, and $MLP$ comprises two linear layers with GELU activation functions. $i$ is the intermediate block identifier, and $M$ is the number of transformer layers.

Throughout the encoding process, the MSDFE module is configured with four instances. Initially, in the first two MSDFE modules, each branch functions independently, extracting features without interacting with each other. In the latter two modules, the outputs of these branches are amalgamated and then transferred to the next "conv flow" stage, facilitating collaborative learning. Specifically, in these later stages, the output from the "trans flow" $z_i \in \mathbb{R}^{(N,K)}$ undergoes dimensional transformation and up-sampling to yield $z \in \mathbb{R}^{H\times W\times \frac{C_{out}}{2}}$, aligning with the dimensions of the "conv flow," followed by merging the outputs from both branches through a concatenation operation. This integrated process is mathematically represented in Equation 6:

$$F_{vi}^{D} = \left[E_{\text{comv}}\left(F_{v(i-1)}^{D}\right), E_{\text{trans}}\left(z_{i-1}\right)\right] \quad i = 1 \ldots 4 \tag{6}$$

In this equation, $F_{v(i-1)}^{D}$ symbolizes the down-sampling output from the previous $(i-1)^{th}$ encoder layer, with $E_{\text{conv}}$ and $E_{\text{trans}}$ signifying the ResBlock and TransBlock, respectively, and $[\cdot]$ represents the concatenation of the two features. In our model configuration, the input image dimensions are set to $H = W = 224$, and the TransBlock's layer configuration $M$ is designated as 4, 3, 3, 2, with a patch size of $P = 16 \times 16$, $P = 768$, resulting in a total patch count of $N = 196$. This approach to MSDFE effectively maps visual information to multimodal spaces, significantly improving the model's performance in subsequent tasks such as medical image segmentation and image captioning. It lays a robust foundation for addressing complex cross-modal challenges.

## 3.3 Multimodal semantic enhancement and captioning module

To leverage the multimodal features $F_{v4}^{D} \in \mathbb{R}^{(h_i,w_i,C_i)}$ extracted during the encoding phase for image segmentation and captioning tasks, we devised a Multimodal Semantic Enhancement and Captioning Module (MSECM). As depicted in Figure 2D, the MSECM consists of two main components: Visual Semantic Enhancement (VSE) and Textual Semantic Enhancement (TSE). VSE adjusts $F_{v4}^{D}$ to generate visual features $F_s \in \mathbb{R}^{(h_i,w_i,C_i)}$ tailored for segmentation tasks. In contrast, TSE refines features $F_g \in \mathbb{R}^{(N,K)}$ for the image captioning task and produces the associated medical text descriptions. The MSECM precisely fine-tunes these features to cater to the specific requirements of each task, ensuring that the extracted features are highly task-specific.

We utilize atrous Chen et al. [57] convolution in the VSE to refine the multimodal features. Atrous convolution extends the receptive field by adjusting the dilation rate, allowing it to capture broader contextual information. Specifically, we use different dilation rates (1, 3, 6, 9) to ensure effective information acquisition across various scales. This multi-scale information capture enhances the specificity of visual features for segmentation tasks, providing a solid foundation for achieving accurate segmentation results. Furthermore, due to the Transformer's strong ability to model the two modalities, we integrate the output of TSE into VSE, forming a comprehensive feature set $F' \in \mathbb{R}^{(h_i,w_i,C_i)}$ for the image segmentation task. This feature set will be employed in the subsequent upsampling decoding process, represented by the following Equations 7–10:

$$F_{va_i} = \sigma\left(BN\left(\text{Conv}_{3\times3}^{d=i}\left(F_{v4}^{D}\right)\right)\right) \quad i = 1, 3, 6, 9 \tag{7}$$

$$F_s = \sigma\left(BN\left(\text{Conv}_{3\times3}\left[F_{va_1}, F_{va_3}, F_{va_6}, F_{va_9}\right]\right)\right) \tag{8}$$

$$F_g = \text{Transformer}\left(F_{v4}^{D}\right) \tag{9}$$

$$F' = \left[F_s, \text{Conv}_{3\times3}\left(\text{Re}\left(F_g\right)\right)\right] \tag{10}$$

In these equations, $\text{Conv}_{3\times3}^{d=i}$ symbolizes atrous convolution, $d = i$ indicates the dilation rate, $\text{Transformer}(\cdot)$ is shorthand for Transformer operation, and $\text{Re}(\cdot)$ represents the reshape operation.

In the TSE, as illustrated in Equation 9, the Transformer module is utilized to optimize the multimodal features, with its self-attention mechanism enabling extensive context capture from within the image. This enhances feature coherence and provides a solid foundation for generating text closely related to the image. We employ a lightweight Long Short-Term Memory (LSTM) network Hochreiter and Schmidhuber [58] as the caption generator for the subsequent generation of medical image captions. This network comprises several interconnected LSTM units, enabling it to effectively process sequential data, which is crucial for generating coherent and informative medical texts. To quantitatively assess the accuracy of the generated texts, we use the cross-entropy loss function $L_{gen}$ to guide the LSTM network's training. The loss function is defined in Equation 11:

$$L_{\text{gen}} = -\sum_{t=1}^{N_T} \log\left(p_t\left(y_t \mid y_1, y_2, \ldots, y_{t-1}; \theta\right)\right) \tag{11}$$

In this formula, $p_t(y_t \mid y_1, y_2, \ldots, y_{t-1}; \theta)$ signifies the probability that the model predicts the current word $y_t$, contingent upon the antecedent words and the model parameters $\theta$. This approach ensures a high degree of alignment between the accuracy of the generated text and actual texts. Subsequently, the generated texts are tokenized and converted into embeddings via a trainable embedding layer, resulting in the linguistic feature $L \in \mathbb{R}^{(T,C_L)}$. This feature is further refined by subsequent modules, specifically tailored for applications in the decoding and analysis processes.

## 3.4 Scale-aware textual attention module

To mitigate the impact of variances between model-generated texts and labeled descriptions in a minority of samples, which may compromise the model's segmentation performance, this research has integrated a Scale-aware Textual Attention Module (SATaM). This module exploits multimodal features $F_{vi}^{D} \in \mathbb{R}^{(H_i,W_i,C_i)}$ extracted at different stages of encoding to enhance the quality of linguistic features $L$. Multimodal features $F_{vi}^{D}(\text{i} = 1, 2, 3, 4)$ from varying encoding layers encapsulate distinct information: superficial layers provide comprehensive, sentence-level insights, while deeper layers deliver granular details, such as lesion-specific word-level information. Both levels are instrumental in guiding the development of linguistic features. Furthermore, SATaM additionally incorporates a semantic consistency loss function (SCloss) to enhance further the attention of linguistic features on key lesion areas. SATaM is

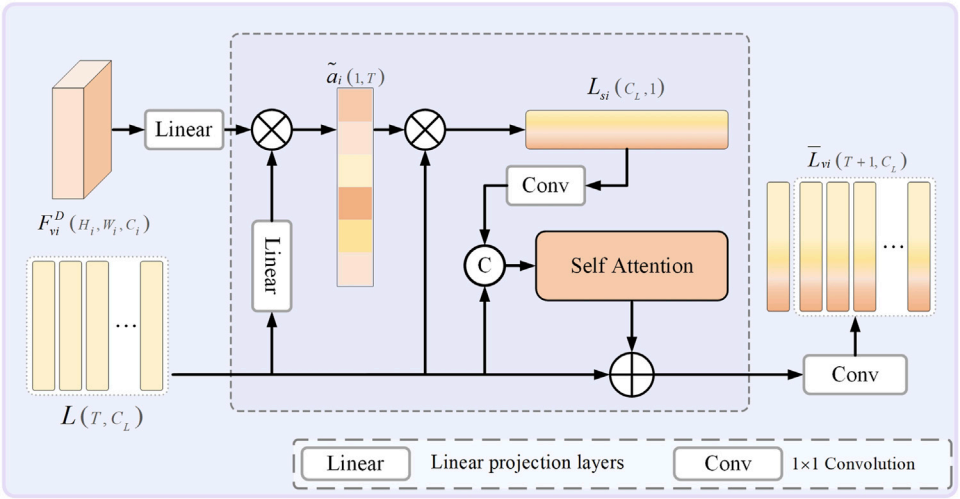



FIGURE 3
Architectures of the scale-aware textual attention module.

designed to allocate higher attention weights to lexical or sentence features while minimizing focus on irrelevant or misleading information. This approach ensures the emphasized features maintain a solid semantic correlation with the visual content. Figure 3 illustrates the architecture of the SATaM. Initially, $F_{vi}^{D}$ and $L$ are mapped through a fully connected layer to a unified subspace, where a cross-modal attention mechanism is applied. This generates an attention map $A_i \in \mathbb{R}^{HW \times T}$, delineating the correlations between T words and every pixel in the image. Subsequently, the map $A_i$ undergoes summation across the $HW$ dimensions and is normalized, resulting in the attention matrix $\tilde{a}_i \in \mathbb{R}^{T}$. This process is graphically represented in the following Equations 12–14:

$$A_i = (\omega_v F_{vi})(\omega_l L) \tag{12}$$

$$a_i = \sum_{j=1}^{HW} A_i^j \tag{13}$$

$$\tilde{a}_i^t = \frac{\exp(a_i^t / \|a_i\|_2)}{\sum_{k=1}^{T} \exp(a_i^k / \|a_i\|_2)} \tag{14}$$

Herein, $\omega_v$ and $\omega_l$ are projection parameters, $\|\cdot\|_2$ denotes the L2-norm, $A_i^j \in \mathbb{R}^T$ the feature relevance between $T$ words and the $jth$ pixel. The term $\tilde{a}_i^t \in \mathbb{R}^T$ indicates the significance of the $t$-th word about the current visual features. Hence, we employ $\tilde{a}_i$ to linearly recombine $L$ across the word dimension, deriving an adaptive, scale-aware sentence features $L_{si} \in \mathbb{R}^{C_L}$. This feature dynamically adjusts its representation in response to visual content of varying scales, enhancing its ability to encompass and articulate overall visual information. Expanding upon this, $L_{si}$ is concatenated with $L$ to forge a novel $T+1$ dimensional linguistic feature $L_i' \in \mathbb{R}^{(T+1, C_L)}$. This improved feature is then processed through a self-attention mechanism, and subsequently, it is combined with $L$ to produce $L_{vi} \in \mathbb{R}^{(T+1, C_L)}$. This operation aims to enrich the original linguistic features of $L$ with visual context provided by $L_{si}$ while preserving the integrity of $L$'s textual structure. The steps of this procedure are detailed in the following Equations 15, 16:

$$L_i' = [\mathrm{Conv}_{1\times1}(L_s), L] \tag{15}$$

$$L_{vi} = \mathrm{Conv}_{1\times1}(Self(L_i') + L) \tag{16}$$

Here, $Self(\cdot)$ refers to the self-attention mechanism. Finally, we remove the token that represents $L_{si}$ from $L_{vi}$, resulting in $\bar{L}_{vi} \in \mathbb{R}^{(T, C_L)}$, which retains the contextual understanding of the original text structure and incorporates scale-level visual information. Thus, $\bar{L}_{vi}$ is utilized as the input for textual information in the decoding phase.

Furthermore, before each skip connection within the model, the SATaM produces four adaptively scale-aware sentence features, $L_{si}\,(i = 1, 2, 3, 4)$. These features are designed to concentrate on lesion areas consistently. To ensure this consistent focus, this study further introduces a SC (Semantic Consistency) Loss comprising three Mean Squared Error (MSE) loss functions: $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. These functions are designed to minimize differences between the sentence-level features $L_{si}$ at various stages, enhancing their focus consistency. The implementation includes the following Equations 17–20:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} \|L_{s1} - L_{s2}\|^2 \tag{17}$$

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N} \|L_{s1} - L_{s3}\|^2 \tag{18}$$

$$\mathcal{L}_3 = \frac{1}{N}\sum_{i=1}^{N} \|L_{s1} - L_{s4}\|^2 \tag{19}$$

$$\mathcal{L}_{con} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 \tag{20}$$

The introduction of SCloss ensures that each SATaM can effectively share insights and impose constraints on one another. This mechanism enables the linguistic features $\bar{L}_{vi}$, guided by $L_{si}$, to target lesion areas that critically affect the segmentation more precisely. Consequently, the interaction of these two mechanisms provides linguistic features relevant to the visual content, complementing the segmentation process during the decoding

stage and significantly enhancing the overall quality of the segmentation results.

## 3.5 Language-aware visual decoder

To optimize the decoding phase of segmentation, we have implemented a Language-Aware Visual Decoder (LAVD). This module is specifically designed to enable more effective integration of features, thereby facilitating the subsequent up-sampling steps. As shown in Figure 2C, the designated input features consist of the decoding features $F_{vi}^{U} \in \mathbb{R}^{(H_i,W_i,C_i)}$ from the preceding stage, the linguistic features $\bar{L}_{vi}$, and the features $F_{vi}^{D}$ from the encoding phase, which serve as skip connections. The decoder aggregates $\bar{L}_{vi}$ along the pixel dimension, creating feature vectors specific to the image pixel positions, which gather the language information most relevant to the current local area. This culminates in spatial attention maps $F_{Ai} \in \mathbb{R}^{(H_i,W_i,C_i)}$, Concretely, we obtain $F_{Ai}$ from the following Equations 21–24:

$$V_{Q_i} = UP\left(\omega_{qi}\left(F_{v_{i-1}}^{U}\right)\right) \tag{21}$$

$$L_{Ki} = \omega_{ki}\left(\bar{L}_{vi}\right) \tag{22}$$

$$L_{Vi} = \omega_{vi}\left(\bar{L}_{vi}\right) \tag{23}$$

$$F_{Ai} = \text{softmax}\left(\frac{V_{Qi}L_{Ki}}{\sqrt{d_i}}\right)L_{Vi} \tag{24}$$

Within this framework, $\omega_{qi}$, $\omega_{ki}$ and $\omega_{vi}$ denote the mappings from linear layers, with $UP(\cdot)$ denoting up-sampling. Using the visual feature $F_{vi}^{U}$ as query and linguistic features $\bar{L}_{vi}$ as both keys and value, the module accomplishes scaled dot-product attention Vaswani et al. [56]. Finally, the acquired $F_{Ai}$ is concatenated with the multimodal features from the encoding phase $F_{vi}^{D}$ and then inputted into the for further learning, as detailed in the following Equation 25:

$$F_{vi}^{U} = \text{Res}\left(\text{Res}\left[F_{Ai}, F_{vi}^{D}\right]\right) \quad i = 1, 2, 3, 4 \tag{25}$$

In our approach, the LAVD is set to 4, and after four iterations of up-sampling, $F_{vi}^{U}$ yields the final segmentation mask of the lesion area.

## 3.6 Overall loss functions

The overall training loss is divided into two main components: segmentation loss $\mathcal{L}_{seg}$ and caption generation loss $\mathcal{L}_{gen}$. For the segmentation part, we have chosen two commonly used losses in medical image segmentation, $\mathcal{L}_{ce}$ and $\mathcal{L}_{dice}$ as well as the semantic consistency loss $\mathcal{L}_{con}$ introduced in this study. These are defined in the following Equations 26–28:

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(p_i\right) \tag{26}$$

$$\mathcal{L}_{\text{dice}} = 1 - \frac{2\sum_{i=1}^{N} p_i y_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} y_i^2} \tag{27}$$

$$\mathcal{L}_{\text{seg}} = \frac{1}{2}\mathcal{L}_{ce} + \frac{1}{2}\mathcal{L}_{dice} + \lambda_c\mathcal{L}_{con} \tag{28}$$

The overall loss function is formulated in Equation 29:

$$\mathcal{L}_{\text{totle}} = \alpha\mathcal{L}_{\text{seg}} + \beta\mathcal{L}_{\text{gen}} \tag{29}$$

Within this construct, $p_i$ and $y_i$ respectively represent the binary segmentation prediction probability for the $i$-th pixel of each input image and the corresponding label classification. N represents the number of pixels. $\lambda_c$, $\alpha$ and $\beta$ signify the hyperparameters applied for weighting various losses. Through the incorporation of $\mathcal{L}_{seg}$ and $\mathcal{L}_{gen}$, Cap2Seg effectively narrows the gap between segmentation maps and labels while generating high-quality medical text annotations, thereby enabling the model to utilize linguistic insights to enhance the segmentation process.

# 4 Experimental

This section comprehensively evaluates our Cap2Seg network using the QaTa-COV19 and MosMedData + datasets. Each experiment is meticulously described, and the results are rigorously analyzed.

## 4.1 Implementation details

This study's methodology was executed on an NVIDIA RTX 4080 using PyTorch. The optimization of model parameters was carried out with an AdamW optimizer that includes a weight decay of 0.0001. Following Li et al. [14], the initial learning rates were configured at 3e-4 for the QaTa-COV19 dataset and 1e-3 for the MosMedData + dataset; due to the differing data sizes of the datasets, batch sizes were specifically configured at 4 for the QaTa-COV19 dataset and 8 for the MosMedData + dataset. The hyperparameters $\alpha$, $\beta$, and $\lambda_c$ were established at 5.0, 2.0, and 0.5 values, respectively. For performance evaluation, we utilized the Dice Thomas et al. [59] coefficient and Mean Intersection over Union (mIoU) Ouyang et al. [60] to assess our model's effectiveness relative to other state-of-the-art methods. These evaluations are computed using the following Equations 30, 31:

$$\text{DSC}_{(A,B)} = \frac{2 \times |A \bigcap B|}{A + B} \tag{30}$$

$$mIoU_{(A,B)} = \frac{1}{N}\sum_{i=1}^{N}\frac{|A \bigcap B|}{|A \bigcup B|} \tag{31}$$

Here, A and B denote the labels and segmentation predictions, respectively.

## 4.2 Datasets

The study utilized two primary public datasets: QaTa-COV19 Degerli et al. [61] and MosMedData + Morozov et al. [62]. The QaTa-COV19 dataset comprises 9,258 chest X-ray images of COVID-19, each with a 224 × 224 pixels resolution. Of these, 5,716 were designated for training, 1,429 for validation, and 2,113 for testing. The MosMedData + dataset contains $2{,}729^{-\Delta\Delta CT}$ scans depicting lung infections, with each image having a resolution of 512 × 512 pixels. It includes 2,183 images for training, 273 for validation, and another 273 for testing purposes. Notably, the original datasets did not include

TABLE 1 Compares the state-of-the-art segmentation methods on the MOSMEDDATA + dataset. GRAY-SHADED methods exclude text input, while others include text input.

| Method | w/o Text | | Generated Text | | Ground Truth Text | |
|---|---|---|---|---|---|---|
| | mIoU[%] | Dice[%] | mIoU[%] | Dice[%] | mIoU[%] | Dice[%] |
| U-Net | 50.73 | 64.60 | – | – | – | – |
| Att-Unet | 52.82 | 66.34 | – | – | – | – |
| UNet++ | 58.39 | 71.75 | – | – | – | – |
| TransUNet | 58.44 | 71.24 | – | – | – | – |
| Swin-Unet | 50.19 | 63.29 | – | – | – | – |
| SCOAT-Net | 56.87 | 70.51 | – | – | – | – |
| COPLE-Net | 60.93 | 74.08 | – | – | – | – |
| ConTEXTualNet | 56.81 | 70.60 | 56.03 | 70.08 | 58.19 | 71.66 |
| LAVT | 56.52 | 70.23 | 55.43 | 69.86 | 60.41 | 73.29 |
| TGANet | 60.18 | 73.30 | 59.28 | 71.81 | 59.28 | 71.81 |
| LViT-T | 60.40 | 72.58 | 59.86 | 73.41 | 61.33 | 74.57 |
| Cap2Seg(Ours) | **63.02** | **75.87** | **63.02** | **75.87** | – | – |

Bold values represent the best performance.

medical text annotations; these were added subsequently by the LVIT Li et al. [14], which provided detailed descriptions of the lesions in terms of their areas, quantities, and locations. Such as **"bilateral lung infection, two infection zones, upper left lung and upper right lung."** indicating bilateral lung infections with two infection zones in the upper left and upper right lungs, and **"unilateral lung infection, one infection zone, lower left lung."** indicating a single-sided lung infection with the infection zone in the lower left lung. Each lesion image corresponds to a medical text annotation, with more detailed textual annotation information available in Li et al. [14].

## 4.3 Results and analysis

We first validated the effectiveness of our method on the MosMedData + dataset and compared it with existing methods under three different conditions. The first condition is that no text modality is used as auxiliary input during inference, corresponding to the "w/o Text" column in the table. The second condition involves using generated medical text annotations to assist segmentation during inference, as shown in the "Generated Text" column in the table. These annotations are generated by Cap2Seg at its optimal performance and are used as inputs for other models. A detailed qualitative evaluation of these generated texts is provided in Section 4.4. The third condition is that real labeled medical text annotations assist segmentation during inference, corresponding to the "Ground Truth Text" column in the table. Since our model does not use any text input during inference and the model generates the auxiliary text, our method falls under the first two conditions. Therefore, we perform inference only under these two conditions and compare it with existing methods. We compared our method with mainstream text-guided image segmentation methods Yang et al. [32]; Li et al. [14]; Huemann et al. [16]; Tomar et al. [17] and some state-of-the-art segmentation methods Ronneberger et al. [9]; Zhou et al. [10]; Oktay et al. [63]; Katore and Thanekar [64]; Chen et al. [65]; Cao et al. [66]; Zhao et al. [67]. The corresponding comparison results are listed in Table 1, with the best results highlighted in bold.

Our findings reveal that Cap2Seg substantially exceeded the performance of existing approaches in the three conditions outlined above. It is important to note that when using generated annotations with discrepancies from real text annotations for segmentation assistance, Li et al. [14]; Huemann et al. [16]; Tomar et al. [17] that did not account for this issue generally saw reduced performance. Nevertheless, Cap2Seg effectively addressed and mitigated this issue. Specifically, using generated medical text annotations, Cap2Seg increased the mIoU score by 3.66% and the Dice score by 2.71% compared to the suboptimal LViT. Even against LViT utilizing real medical text annotations, Cap2Seg still improved the mIoU score by 1.69% and the Dice score by 1.3%. These results suggest that Cap2Seg adeptly learned lesion-related visual cues, minimized its dependency on potentially misleading information, and underscored its superior capability.

In further evaluations conducted on the QaTa-COV19 dataset, the quantitative comparisons of our Cap2Seg are detailed in Table 2. Specifically, Cap2Seg achieved a mean Intersection over Union (mIoU) of 71.61% and a Dice coefficient of 81.32%. Cap2Seg achieved the best or near-best results in all three evaluated scenarios, demonstrating its significant superiority over the previously discussed state-of-the-art methods.

## 4.4 Visual comparison of segmentation results

We have conducted visual qualitative assessments of our Cap2Seg method on the MosMedData+ and QaTa-COV19 datasets,

TABLE 2 Compares the state-of-the-art segmentation methods on the QaTa-COV19 dataset. GRAY-SHADED methods exclude text input, while others include text input.

| Method | w/o Text | | Generated Text | | Ground Truth Text | |
|---|---|---|---|---|---|---|
| | mIoU[%] | Dice[%] | mIoU[%] | Dice[%] | mIoU[%] | Dice[%] |
| U-Net | 69.46 | 79.02 | – | – | – | – |
| Att-Unet | 70.04 | 79.31 | – | – | – | – |
| UNet++ | 70.25 | 79.62 | – | – | – | – |
| TransUNet | 69.13 | 78.63 | – | – | – | – |
| Swin-Unet | 68.34 | 78.07 | – | – | – | – |
| SCOAT-Net | 69.85 | 79.59 | – | – | – | – |
| COPLE-Net | 70.81 | 80.12 | – | – | – | – |
| ConTEXTualNet | 68.67 | 78.15 | 68.74 | 78.49 | 70.16 | 79.60 |
| LAVT | 61.21 | 72.61 | 68.10 | 78.04 | 69.89 | 79.28 |
| TGANet | 69.09 | 78.46 | 70.75 | 79.87 | 70.75 | 79.87 |
| LViT-T | 71.37 | 81.12 | 69.19 | 78.17 | **75.11** | **83.66** |
| Cap2Seg(Ours) | **71.61** | **81.32** | **71.61** | **81.32** | – | – |

Bold values represent the best performance.

benchmarking it against current methodologies. As illustrated in Figure 4, segmentation inaccuracies are noticeable in the outputs from CopleNet Katore and Thanekar [64], ConTEXTualNet Huemann et al. [16], TGANet Tomar et al. [17], and LViT Li et al. [14] across the first, third and fourth rows, where these methods exhibit erroneous segmentation zones. In contrast, our approach effectively delineates the primary regions of lesions. Moreover, the sixth row demonstrates that while existing methods struggle with identifying lesion peripheries and finer details, Cap2Seg excels in recognizing these critical features, showcasing our network's enhanced capability to capture lesion-specific areas accurately. The visual evidence indicates that our method achieves comparable or superior segmentation results relative to other models.

## 4.5 Ablation study

The proposed method is structured around three principal components: MSDFE, MSECM, and SATaM, with SATaM integrating SCloss, a feature proven effective in our analysis. The following ablation experiments were conducted to evaluate the efficacy of each component individually. MSDFE, MSECM, and SATaM were initially removed from our model to create a baseline. These components were then incrementally reintroduced to assess their contributions. This methodology was validated using the results from the MosMedData + dataset, summarized in Table 3, which indicated that the gradual reintroduction of these modules allowed our complete model to achieve an optimal mIoU score of 63.02% and a Dice score of 75.87%. The segmentation results for different configurations, illustrated in Figure 5, reveal that our full model achieves exceptional segmentation precision, especially in the location and border areas of lesions.

### 4.5.1 Effectiveness of MSDFE

In this study, the MSDFE module extracts a comprehensive set of multimodal features, effectively tackling complex cross-modal challenges. To ascertain the efficacy of this approach, we analyzed the segmentation performance differences between the "Baseline" and "Baseline*". According to the results in Table 3, "Baseline*" reached mIoU and Dice scores of 60.96% and 74.18% respectively, showing improvements of 1.72% and 1.35% over "Baseline. "The visual segmentation outcomes depicted in Figure 5 corroborate these findings, showing that incorporating the MSDFE module notably decreases segmentation errors, particularly within lesion regions.

Furthermore, the study delved into the effects of interactions between two designated sub-modules, ResBlock and TransBlock, within various MSDFE modules during the encoding stage. Table 4 reveals that initiating these interactions from the third MSDFE module optimizes model performance. This evidence collectively emphasizes the MSDFE module's pivotal role in enhancing feature recognition capabilities in lesion areas and elevating overall segmentation accuracy.

### 4.5.2 Effectiveness of MSECM

The proposed MSECM module finely tunes the encoded multimodal features to enhance their task specificity. This adjustment results in two more features aligned with the intended tasks. To evaluate the effectiveness of MSECM, we analyzed the data presented in Table 3. The introduction of MSECM improved the mIoU and Dice scores of "Baseline* + MSECM" by 0.74% and 0.68%, respectively, compared to "Baseline*." These findings demonstrate that integrating MSECM into our network markedly improves mIoU and Dice scores, confirming its beneficial impact on the model's performance.

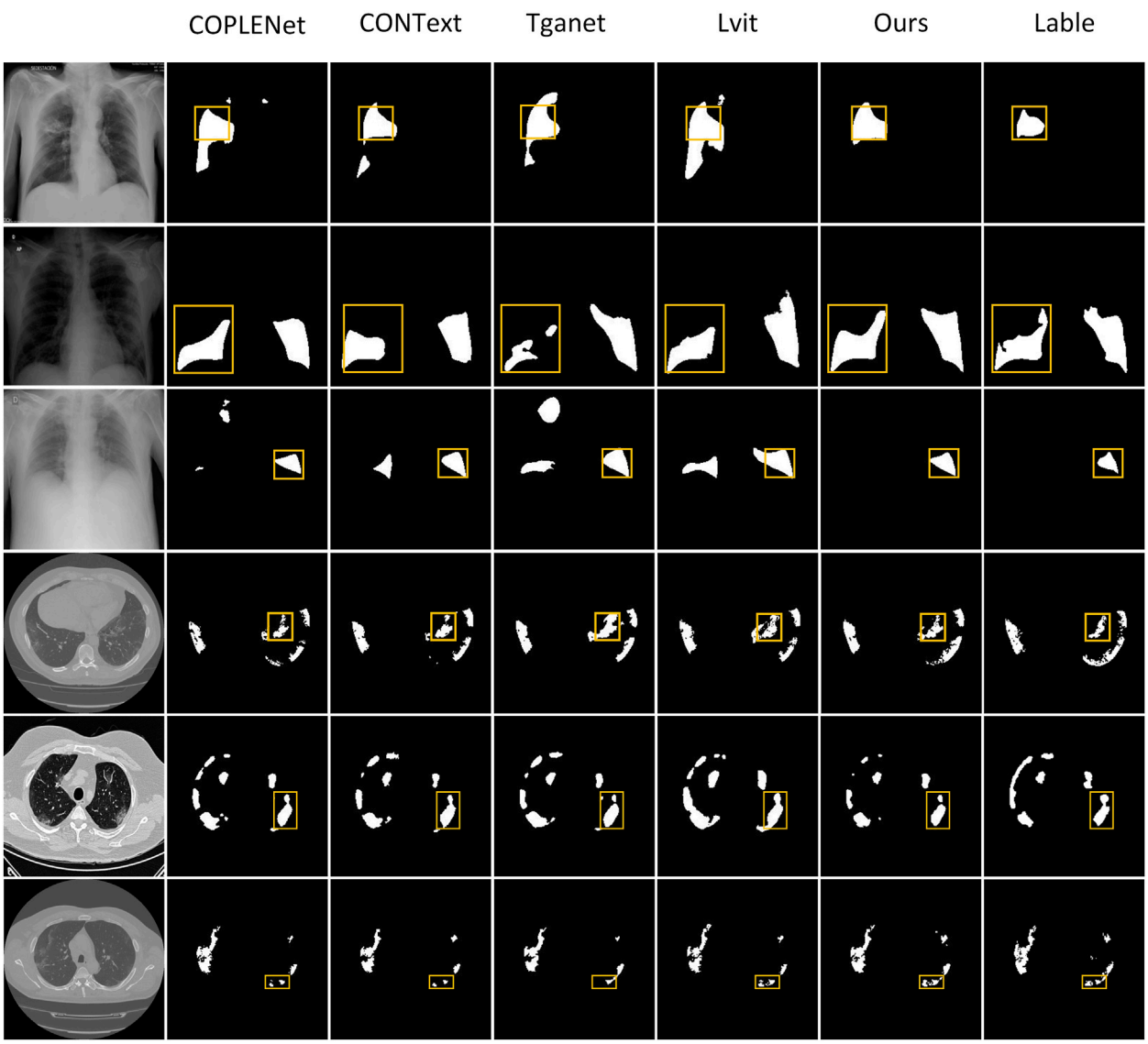



FIGURE 4
Visual comparison of the proposed method with different segmentation methods.

TABLE 3 Ablation study on the MOSMEDDATA + dataset. "Baseline" represents the utilization OF CNN as the encoder. "Baseline*" indicates the employment of the MSDFE introduced in this paper as the encoder. "SATaM\SCloss" indicates the removal of SCloss from SATaM.

| Methods | mIoU [%] | Dice [%] |
|---|---|---|
| Baseline | 59.24 | 72.83 |
| Baseline* | 60.96 | 74.18 |
| Baseline* + MSECM | 61.70 | 74.86 |
| Baseline* + SATaM | 61.56 | 74.59 |
| Baseline* + MSECM + SATaM\SCloss | 62.23 | 74.92 |
| Baseline* + MSECM + SATaM | 63.02 | 75.87 |

### 4.5.3 Effectiveness of SATaM

SATaM assigns attention weights related to visual information to linguistic features, thus enhancing their focus on crucial lesion areas and ensuring a tight linkage between linguistic characteristics and these areas. Consequently, this reduces the model's attention to irrelevant or misleading textual features, enhancing its segmentation capabilities. The significant improvements in segmentation performance with the inclusion of SATaM are evident in Figure 5, validating its utility. Table 3 shows that "Baseline* + MSECM + SATaM" achieved increases of 1.32% and 1.01% in mIoU and Dice scores, respectively, compared to "Baseline* + MSECM." The impact of Semantic Consistency Loss (SCloss) within SATaM was also assessed. The removal of SCloss led to diminished performance in the "Baseline* + MSECM + SATaM\SCloss" configuration, highlighting

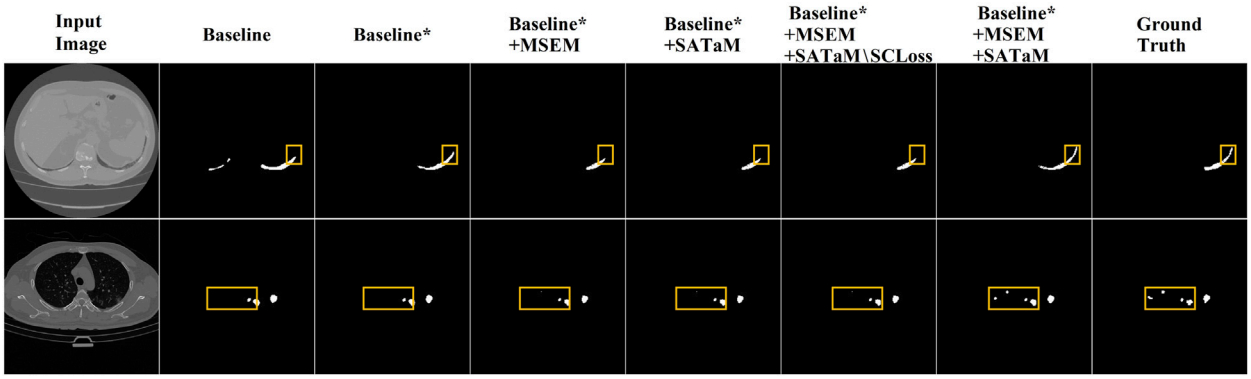



FIGURE 5
Visual comparison of the effectiveness of different components.

TABLE 4 Impact of interaction between two submodules in different MSDFE modules during the encoding stage. '✓' indicates interaction within the current MSDFE.

| MSDFE1 | MSDFE2 | MSDFE3 | MSDFE4 | mIoU [%] | Dice [%] |
|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | 61.89 | 74.78 |
| | ✓ | ✓ | ✓ | 61.43 | 74.54 |
| | | ✓ | ✓ | **63.02** | **75.87** |
| | | | ✓ | 61.02 | 74.06 |

Bold values represent the best performance.

SCloss's pivotal role within the SATaM framework. These findings confirm that SATaM substantially boosts the model's segmentation accuracy, resulting in more precise and consistent predictions.

## 5 Conclusion

This paper proposes Cap2Seg, a network that combines image segmentation and caption generation tasks. The introduction of the MSECM effectively coordinates both tasks, enhancing multi-task learning efficiency. The SATaM reduces the model's reliance on irrelevant or misleading textual information, while the LAVD effectively fuses textual features with visual features. By generating text to guide the segmentation task, Cap2Seg fully leverages the potential of textual annotations, thereby improving the quality and accuracy of COVID-19 image segmentation. It eliminates the dependency on image-text pairs and provides additional textual references for clinical diagnosis. Extensive experimental results confirm the proposed method's effectiveness and superiority over existing approaches. Ablation experiments also validate the efficacy of each core component of the proposed model. However, it is essential to acknowledge that although our method has achieved satisfactory results in image segmentation, we still face challenges in accurately generating specific keywords in a few samples, affecting the segmentation performance when dealing with complex lesion images. Additionally, due to the scarcity of paired medical image and text datasets, our method has only been validated on two COVID-19 datasets. Currently, we have not fully resolved the challenge. In future research, we will expand our study to more types of disease datasets. Meanwhile, we plan to optimize the caption generation module to improve the model's ability to capture key lesion areas and explore more effective feature interaction and fusion strategies. These improvements and extensions will help further enhance the practicality and accuracy of our method in segmentation tasks.

## Data availability statement

Publicly available datasets were analyzed in this study. The QaTa-COV19 dataset can be found at https://www.kaggle.com/datasets/aysendegerli/qatacov19-dataset, and the MosMedData+ dataset can be found at https://www.kaggle.com/datasets/maedemaftouni/covid19-ct-scan-lesion-segmentation-dataset.

## Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

## Author contributions

WZ: Methodology, Software, Writing–original draft, Writing–review and editing. FL: Funding acquisition, Resources, Supervision, Writing–review and editing. YD: Conceptualization, Formal Analysis, Writing–review and editing. PF: Data curation, Visualization, Writing–review and editing. ZC: Validation, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was supported by the National Natural Science Foundation of China (Nos. 62362045), the Basic Research Project of Yunnan Province (Nos. 202401AT070412).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note



## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2024.1439122/full#supplementary-material

## References

1. Benvenuto D, Giovanetti M, Salemi M, Prosperi MCF, Flora CD, Alcantara LCJ, et al. The global spread of 2019-ncov: a molecular evolutionary analysis. *Pathog Glob Health* (2020) 114:64–7. doi:10.1080/20477724.2020.1725339

2. World Health Organization and others Coronavirus disease 2019 (COVID-19): situation report, 73, (2020). World Health Organization.

3. Chan JF-W, Yuan S, Kok K-H, To KK-W, Chu H, Yang J, et al. A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster. *Lancet (London, England)* (2020) 395:514–23. doi:10.1016/s0140-6736(20)30154-9

4. Raoof S, Volpi A. The role of chest imaging in patient management during the COVID-19 pandemic: a multinational consensus statement from the Fleischner Society. *Radiology* (2020). 296(1), 172–180. Radiological Society of North America.

5. Shi W, Peng X, Liu T, Cheng Z, Lu H, Yang S, et al. A deep learning-based quantitative computed tomography model for predicting the severity of covid-19: a retrospective study of 196 patients. *Ann Translational Med* (2020) 9:216. doi:10.21037/atm-20-2464

6. Zhu Z, He X, Qi G, Li Y, Cong B, Liu Y. Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal mri. *Inf Fusion* (2022) 91:376–87. doi:10.1016/j.inffus.2022.10.022

7. Liu Y, Shi Y, Mu F, Cheng J, Chen X. Glioma segmentation-oriented multi-modal mr image fusion with adversarial learning. *IEEE CAA J Autom Sinica* (2022) 9:1528–31. doi:10.1109/jas.2022.105770

8. Zhu Z, Wang Z, Qi G, Mazur N, Yang P, Liu Y. Brain tumor segmentation in mri with multi-modality spatial information enhancement and boundary shape correction. *Pattern Recognition* (2024) 153:110553. doi:10.1016/j.patcog.2024.110553

9. Ronneberger O., Fischer P., Brox T. (2015). U-net: convolutional networks for biomedical image segmentation, 234, 41. doi:10.1007/978-3-319-24574-4_28

10. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J. Unet++: a nested u-net architecture for medical image segmentation. *Deep Learning in medical image Analysis and multimodal Learning for clinical decision support: 4th international workshop, dlmia 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with miccai 2018, Proceedings 4* (2018) 3–11. Springer.

11. Wang G, Liu X, Li C, Xu Z, Ruan J, Zhu H, et al. A noise-robust framework for automatic segmentation of covid-19 pneumonia lesions from ct images. *IEEE Trans Med Imaging* (2020) 39:2653–63. doi:10.1109/tmi.2020.3000314

12. Fan D-P, Zhou T, Ji G-P, Zhou Y, Chen G, Fu H, et al. Inf-net: automatic covid-19 lung infection segmentation from ct images. *IEEE Trans Med Imaging* (2020) 39:2626–37. doi:10.1109/tmi.2020.2996645

13. Monajatipoor M, Rouhsedaghat M, Li LH, Chien A, Kuo C-CJ, Scalzo F, et al. Berthop: an effective vision-and-language model for chest x-ray disease diagnosis. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW); Oct. 17 2021; China (2021). p. 3327–36.

14. Li Z, Li Y, Li Q, Zhang Y, Wang P, Guo D, et al. Lvit: language meets vision transformer in medical image segmentation. *IEEE Trans Med Imaging* (2022) 43:96–107. doi:10.1109/tmi.2023.3291719

15. Chen W, Liu J, Yuan Y. Bi-vlgm: Bi-level class-severity-aware vision-language graph matching for text guided medical image segmentation. (2023). *ArXiv abs/2305.12231*

16. Huemann Z, T Xin, Hu J, Bradshaw TJ. Contextual net: a multimodal vision-language model for segmentation of pneumothorax. *Journal of Imaging Informatics in Medicine*. (2024). Springer. 1–12.

17. Tomar NK, Jha D, Bagci U, Ali S. Tganet: text-guided attention for improved polyp segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*. (2022) 151–160. Springer.

18. Wen Y, Chen L, Qiao L, Deng Y, Chen H, Zhang T, et al. Let's find fluorescein: cross-modal dual attention learning for fluorescein leakage segmentation in fundus fluorescein angiography. In: 2021 IEEE International Conference on Multimedia and Expo (ICME); July 5-9, 2021; China, 67 (2021). p. 1–6. doi:10.1109/icme51207.2021.9428108

19. Li Y, Luo L, Lin H, Chen H, Heng P-A. Dual-consistency semi-supervised learning with uncertainty quantification for covid-19 lesion segmentation from ct images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (2021). p. 199–209. doi:10.1007/978-3-030-87196-3_19

20. Yu X, Wang J, Hong Q-Q, Teku R, Wang S, Zhang Y. Transfer learning for medical images analyses: a survey. *Neurocomputing* (2022) 489:230–54. doi:10.1016/j.neucom.2021.08.159

21. Wen Y, Chen L, Qiao L, Deng Y, Chen H, Zhang T, et al. Fleak-seg: automated fundus fluorescein leakage segmentation via cross-modal attention learning. *IEEE MultiMedia* (2022) 29:114–23. doi:10.1109/mmul.2022.3142986

22. Hu R, Rohrbach M, Darrell T (2016). Segmentation from natural language expressions. *ArXiv* abs/1603.06180

23. Liu C, Lin ZL, Shen X, Yang J, Lu X, Yuille AL. Recurrent multimodal interaction for referring image segmentation. In: 2017 IEEE International Conference on Computer Vision (ICCV); 22-29 Oct. 2017; China (2017). p. 1280–9.

24. Li R, Li K, Kuo Y-C, Shu M, Qi X, Shen X, et al. Referring image segmentation via recurrent refinement networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18-23 June 2018; USA (2018). p. 5745–53.

25. Shi X, Chen Z, Wang H, Yeung DY, Wong W-K, chun Woo W. Convolutional lstm network: a machine learning approach for precipitation nowcasting. *Neural Inf Process Syst* (2015, 28).

26. Ye L, Rochan M, Liu Z, Wang Y. Cross-modal self-attention network for referring image segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 15-20 June 2019; China (2019). p. 10494–503.

27. Shi H, Li H, Meng F, Wu Q. Key-word-aware network for referring expression image segmentation. In: European Conference on Computer Vision (2018). p. 38–54. doi:10.1007/978-3-030-01231-1_3

28. Wang X, Girshick RB, Gupta AK, He K. Non-local neural networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18-23 June 2018; China (2017). p. 7794–803.

29. Chen D-J, Jia S, Lo Y-C, Chen H-T, Liu T-L. See-through-text grouping for referring image segmentation. In: 2019 IEEE/CVF international conference on computer vision (ICCV), 18-23 June 2018; China, (2019). p. 7453–62.

30. Hu Z, Feng G, Sun J, Zhang L, Lu H. Bi-directional relationship inferring network for referring image segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 13-19 June 2020; China (2020). p. 4423–32.

31. Chen D-J, Hsieh H-Y, Liu T-L. Referring image segmentation via language-driven attention. In: 2021 IEEE International Conference on Robotics and Automation (ICRA); 2021 May 30; China (2021). p. 13997–4003.

32. Yang Z, Wang J, Tang Y, Chen K, Zhao H, Torr PHS. Lavt: language-aware vision transformer for referring image segmentation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 18-24 June 2022; USA (2021). p. 18134–44.

33. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021).

34. Li B, Weinberger KQ, Belongie SJ, Koltun V, Ranftl R (2022). Language-driven semantic segmentation. *ArXiv* abs/2201.03546

35. Xu J, Mello SD, Liu S, Byeon W, Breuel T, Kautz J, et al. Groupvit: semantic segmentation emerges from text supervision. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 18-24 June 2022; USA (2022). p. 18113–23.

36. Yu L, Lin ZL, Shen X, Yang J, Lu X, Bansal M, et al. Mattnet: modular attention network for referring expression comprehension. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18-23 June 2018; USA (2018). p. 1307–15.

37. Huang S, Hui T, Liu S, Li G, Wei Y, Han J, et al. Referring image segmentation via cross-modal progressive comprehension. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 13-19 June 2020; Germany (2020). p. 10485–94.

38. Hui T, Liu S, Huang S, Li G, Yu S, Zhang F, et al. Linguistic structure guided context modeling for referring image segmentation. In: European Conference on Computer Vision (2020). p. 59–75. doi:10.1007/978-3-030-58607-2_4

39. Ding H, Liu C, Wang S, Jiang X. Vlt: vision-language transformer and query generation for referring segmentation. *IEEE Trans Pattern Anal Machine Intelligence* (2022) 45:7900–16. doi:10.1109/tpami.2022.3217852

40. Huang S-C, Shen L, Lungren MP, Yeung S. Gloria: a multimodal global-local representation learning framework for label-efficient medical image recognition. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 10-17 Oct. 2021; China (2021). p. 3922–31.

41. Zhang Y, Jiang H, Miura Y, Manning CD, Langlotz C. Contrastive learning of medical visual representations from paired images and text. *Machine Learning for Healthcare Conference* (2020), 2–25. PMLR.

42. Dai L, Fang R, Li H, Hou X, Sheng B, Wu Q, et al. Clinical report guided retinal microaneurysm detection with multi-sieving deep learning. *IEEE Trans Med Imaging* (2018) 37:1149–61. doi:10.1109/tmi.2018.2794988

43. Vinyals O, Toshev A, Bengio S, Erhan D. Show and tell: a neural image caption generator. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 7-12 June 2015; China (2014). p. 3156–64.

44. Ghandi T, Pourreza HR, Mahyar H. Deep learning approaches on image captioning: a review. *ACM Comput Surv* (2022) 56:1–39. doi:10.1145/3617592

45. Zhang W, Ying Y, Lu P, Zha H. Learning long- and short-term user literal-preference with multimodal hierarchical transformer network for personalized image caption. In: AAAI Conference on Artificial Intelligence, 34 (2020). p. 9571–8. doi:10.1609/aaai.v34i05.6503

46. Yu J, Li J, Yu Z, Huang Q. Multimodal transformer with multi-view visual representation for image captioning. *IEEE Trans Circuits Syst Video Technology* (2019) 30:4467–80. doi:10.1109/tcsvt.2019.2947482

47. Li Y, Liang X, Hu Z, Xing EP. Knowledge-driven encode, retrieve, paraphrase for medical image report generation. (2019) 33, 6666, 73. doi:10.1609/aaai.v33i01.33016666

48. Hou D, Zhao Z, Liu Y, Chang F, Hu S. Automatic report generation for chest x-ray images via adversarial reinforcement learning. *IEEE Access* (2021) 9:21236–50. doi:10.1109/access.2021.3056175

49. Wang F, Liang X, Xu L, Lin L. Unifying relational sentence generation and retrieval for medical image report composition. *IEEE Trans Cybernetics* (2020) 52:5015–25. doi:10.1109/tcyb.2020.3026098

50. Wu J, Li X, Ding H, Li X, Cheng G, Tong Y, et al. Betrayed by captions: joint caption grounding and generation for open vocabulary instance segmentation. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV); October 2 - 6, 2023; China (2023). p. 21881–91.

51. Sun M, Suo W, Wang P, Zhang Y, Wu Q. A proposal-free one-stage framework for referring expression comprehension and generation via dense cross-attention. *IEEE Trans Multimedia* (2023) 25:2446–58. doi:10.1109/tmm.2022.3147385

52. Zhang Y, Li H, Du J, Qin J, Wang T, Chen Y, et al. 3d multi-attention guided multi-task learning network for automatic gastric tumor segmentation and lymph node classification. *IEEE Trans Med Imaging* (2021) 40:1618–31. doi:10.1109/tmi.2021.3062902

53. Ioffe S, Szegedy L. Batch normalization: accelerating deep network training by reducing internal covariate shift. (2015). *ArXiv* abs/1502.03167

54. Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: International Conference on Machine Learning (2010).

55. Dosovitskiy A. An image is worth 16x16 words: transformers for image recognition at scale. (2020), *ArXiv* abs/2010.11929.

56. Vaswani A, Shazeer NM, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Neural Inf Process Syst* (2017).

57. Chen L-C, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. (2017), *ArXiv abs/1706.05587*.

58. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* (1997) 9:1735–80. doi:10.1162/neco.1997.9.8.1735

59. Thomas E, Jogi PS, Kumar S, Horo A, Niyas S, Vinayagamani S, et al. Multi-res-attention unet: a cnn model for the segmentation of focal cortical dysplasia lesions from magnetic resonance images. *IEEE J Biomed Health Inform* (2020) 25(5), :1724–34.

60. Ouyang T, Yang S, Gou F, Dai Z, Wu J Rethinking u-net from an attention perspective with transformers for osteosarcoma mri image segmentation. *Comput Intelligence Neurosci* (2022) 2022:1–17. doi:10.1155/2022/7973404

61. Degerli A, Kiranyaz S, Chowdhury MEH, Gabbouj M Osegnet: operational segmentation network for covid-19 detection using chest x-ray images. In: 2022 IEEE International Conference on Image Processing (ICIP); 16-19 Oct. 2022; USA (2022). p. 2306–10.

62. Morozov S, Andreychenko AE, Pavlov NA, Vladzymyrskyy A, Ledikhova NV, Gombolevskiy VA, et al. Mosmeddata: chest ct scans with covid-19 related findings. (2020), *ArXiv* abs/2005.06465.

63. Oktay O, Schlemper J, Folgoc LL, Lee MJ, Heinrich MP, Misawa K, et al. Attention u-net: learning where to look for the pancreas. (2018), *ArXiv abs/1804.03999*.

64. Katore MK, Thanekar PS. A noise-resilient framework for automatic covid-19 pneumonia lesions segmentation from ct images. *Int J Adv Res Sci Commun Technology* (2022) 324–30. doi:10.48175/ijarsct-3746

65. Chen J, Lu Y, Yu Q, Luo X, Adeli E, Wang Y, et al. Transunet: transformers make strong encoders for medical image segmentation. (2021), *ArXiv* abs/2102.04306.

66. Cao H, Wang Y, Chen J, Jiang D, Zhang X, Tian Q, et al. Swin-unet: unet-like pure transformer for medical image segmentation. In: *ECCV workshops* (2021)

67. Zhao S, Li Z, Chen Y, Zhao W, Xie X, Liu J, et al. Scoat-net: a novel network for segmenting covid-19 lung opacification from ct images. *Pattern Recognition* (2020) 119:108109. doi:10.1016/j.patcog.2021.108109

68. Papineni K, Roukos S, Ward T, Zhu W-J. Bleu: a method for automatic evaluation of machine translation. *Annu Meet Assoc Comput Linguistics* (2002).

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 29 November 2024
DOI 10.3389/fphy.2024.1489245



OPEN ACCESS

EDITED BY
Haoyu Chen,
University of Oulu, Finland

REVIEWED BY
Caixia Zheng,
Northeast Normal University, China
Zhen Liu,
Hubei Engineering University, China

*CORRESPONDENCE
Rui Feng,
frengrui@yzu.edu.cn

RECEIVED 31 August 2024
ACCEPTED 23 October 2024
PUBLISHED 29 November 2024

CITATION
Gao W, Feng R and Sheng X (2024) Lightweight multi-stage temporal inference network for video crowd counting. *Front. Phys.* 12:1489245. doi: 10.3389/fphy.2024.1489245



# Lightweight multi-stage temporal inference network for video crowd counting

Wei Gao[1,2], Rui Feng[3]* and Xiaochun Sheng[2]

[1]School of Educational Science, Yangzhou University, Yangzhou, China, [2]School of Computer Engineering, Jiangsu University of Technology, Changzhou, China, [3]School of Journalism and Communication, Yangzhou University, Yangzhou, China

Crowd density is an important metric for preventing excessive crowding in a particular area, but it still faces challenges such as perspective distortion, scale variation, and pedestrian occlusion. Existing studies have attempted to model the spatio-temporal dependencies in videos using LSTM and 3D CNNs. However, these methods suffer from large computational costs, excessive parameter redundancy, and loss of temporal information, leading to difficulties in model convergence and limited recognition performance. To address these issues, we propose a lightweight multi-stage temporal inference network (LMSTIN) for video crowd counting. LMSTIN effectively models the spatio-temporal dependencies in video sequences at a fine-grained level, enabling real-time and accurate video crowd counting. Our proposed method achieves significant performance improvements on three public crowd counting datasets.

KEYWORDS

**crowd counting, crowd density, spatio-temporal dependencies, temporal inference, deep learning**

## 1 Introduction

Crowd counting technology has broad application prospects in fields such as video surveillance, traffic control, and emergency management, and it has been widely applied in urban public safety. In recent years, due to the great success of Convolutional Neural Networks (CNNs) in image classification and object detection, many researchers have introduced CNNs into the crowd counting task to learn the mapping from input images to their corresponding density maps. CNNs are highly favored in the field of crowd counting due to their strong feature learning capabilities, leading to the emergence of numerous outstanding works. Although CNNs have significantly improved the performance of crowd counting methods, most efforts focus on learning feature representations from a single image. These image-based methods still face several challenges that need to be overcome. This is mainly because crowd gatherings can occur in any scenario, such as indoors, outdoors, or in the wild, and both individuals and crowds exhibit rich visual variations. These complex variation factors pose challenges to crowd counting methods, such as occlusion and scale variations, as illustrated in Figure 1.

Existing research has shown that the spatio-temporal information in video sequences contains a wealth of valuable deep semantic information. Modeling the temporal sequence of videos can significantly enhance the feature learning capabilities and discrimination


FIGURE 1
Examples of existing challenges in crowd counting. **(A)** Occlusion. **(B)** Complex backround. **(C)** Scale change.

performance of deep networks. Motion information not only helps produce higher-quality density maps by combining feature representations of adjacent frames, but also improves pedestrian discrimination in occluded scenes. Even if pedestrians are occluded in specific frames, the missing information can still be captured from adjacent frames. Recently, some researchers have attempted to use variants of Long Short-Term Memory (LSTM) networks and 3D Convolutional Neural Networks (3DCNNs) to model the spatio-temporal dependencies in videos, implicitly combining spatial and temporal features [1–6]. Although these methods have achieved some promising results, they suffer from high computational complexity, difficulty in training the related parameters, and the inability to effectively extract long-range temporal context information. These problems lead to low training efficiency and excessive redundant parameters, which limit the model's performance. The Temporal Convolutional Network (TCN) is a neural network model specifically designed for processing time series data. Compared to traditional recurrent neural networks (such as LSTM and GRU), TCN offers the advantages of parallel computation, efficient long-term dependency capture, stable gradients, and flexibility in handling time series of varying lengths. Additionally, the crowd density maps produced by existing methods only offer a rough estimate of crowd distribution and fail to accurately capture individual pedestrian positions or detailed crowd patterns. This limitation significantly hinders further crowd analysis and reduces their practical applicability.

To address these problems, we propose a lightweight multi-stage temporal inference network (LMSTIN) for video crowd counting, which consists of three components: a density map generation module, a lightweight feature extraction module, and a refined temporal inference module. The input to LMSTIN is a sequence of consecutive video frames, and the output is the corresponding crowd density maps. The number of people in each frame is obtained by integrating the density map. Specifically, the density map generation module first uses a focal inverse distance transform to convert the input video frames into crowd density maps with accurate pedestrian positions, which are used as ground truth labels for network training. Then, a lightweight feature extraction module is designed to reduce computational cost while maintaining effective spatial feature extraction, thereby improving the overall efficiency of the network. Finally, a refined temporal inference module is constructed to focus on modeling the dependencies along the temporal dimension. It repeatedly refines the important temporal context information through multiple stages of refined temporal inference to learn better video-level semantic features, further improving crowd counting accuracy. Compared to existing video-based crowd counting methods, LMSTIN achieves promising results on three public video crowd counting datasets. Testing shows that our proposed method demonstrates outstanding performance, meeting the requirements of practical applications in terms of both speed and accuracy.

## 2 Related work

In recent years, with the rapid development of deep learning, there have been significant improvements in the performance of crowd counting methods. Both the accuracy and speed of counting in crowded scenes have notably increased. Fu et al. [7] proposed the first crowd counting model based on Convolutional Neural Networks (CNNs). This model removed some similar network connections in the feature maps and cascaded two CNN classifiers, effectively enhancing the speed and accuracy of crowd counting. Wang et al. [8] introduced a deep network based on the AlexNet structure [9] for extremely dense crowd counting. This network added extra negative samples during training, setting their true values to zero, to reduce the interference from complex backgrounds. Zhang et al. [10] proposed a cross-scene counting network called CrowdCNN based on the AlexNet structure. This network alternately trains on two related tasks (crowd density and crowd counting) to achieve locally optimal results and then fine-tunes the model using pre-training. The multi-column CNN network includes multiple columns of convolutions to extract multi-scale features, thus generating high-quality crowd density maps. Zhang et al. [11] were the first to use a multi-column structure for crowd counting, addressing the problem of scale variation in crowd counting. They proposed the Multi-Column Convolutional Neural Network (MCNN), which consists of different columns, each using filters with varying receptive fields to extract multi-scale features adapted to scene changes. Zhang et al. [12] utilized Local Self-attention (LSA) and Global Self-attention (GSA) to capture short-term and long-term dependencies between pixels

and introduced a relation module to fuse LSA and GSA for richer feature representation. Compared to multi-column CNN methods, single-column CNN methods use a deeper single network structure for feature representation, resulting in a simpler network architecture and easier training convergence. Hu et al. [13] proposed a refinement distance compensation method based on a quantum scale perception learning framework to address crowd counting and localization tasks. This method uses a classic CNN architecture and calculates crowd features through qubit rotation and Pauli operators to generate the final density map. Liu et al. [14] proposed a deformable convolutional network with attention, ADCrowdNet, which consists of an Attention Map Generator (AMG) and a Density Map Estimator (DME). AMG estimates the crowd region and its density in the image, while DME uses multi-scale deformable convolutional layers to generate the crowd density map. Given the great success of Vision Transformers (ViT) in image processing, methods based on ViT have also begun to appear in the field of crowd counting. Liang et al. [15] proposed a crowd counting model called TransCrowd, which was the first to introduce ViT into the crowd counting task, redefining the weakly supervised crowd counting problem from the perspective of image patch sequences based on ViT. TransCrowd effectively utilizes ViT's self-attention mechanism to extract semantic information about crowds, achieving significant crowd counting results. Li et al. [16] improved the ViT model by proposing a new network called CCTrans. This network first uses a pyramid vision transformer backbone to capture global crowd information, then merges low-level and high-level features through a pyramid feature aggregation module, and finally predicts the crowd density map with an efficient multi-scale dilated convolution. Bai et al. [17] proposed an end-to-end crowd counting method called CounTr, which consists of a ViT-based hierarchical encoder-decoder architecture. The encoder inputs image patch sequences to obtain multi-scale features, while the decoder merges features from different layers and aggregates both local and global contextual feature representations.

Deep learning-based crowd counting methods have demonstrated significant capabilities in feature learning for image-level tasks due to the powerful feature learning capabilities of deep neural networks. However, their performance still faces bottlenecks. Recently, many researchers have suggested that modeling the spatio-temporal information contained in video sequences could further overcome these performance limitations. However, research on this approach for crowd counting tasks remains relatively scarce.

# 3 Method

When addressing challenges such as significant scale variation and frequent occlusions in crowd counting, a key issue is how to extract contextual information across video frames and effectively model spatio-temporal dependencies, all while maintaining real-time algorithmic performance. To tackle this, we propose a novel framework, LMSTIN, which achieves fast and accurate video crowd counting by constructing finer-grained spatio-temporal dependencies. Figure 2 presents the overall structure of LMSTIN. LMSTIN consists of three components: a density map generation module, a lightweight feature extraction module, and a refined temporal inference module. Specifically, LMSTIN first employs a density map generation module (DMGM) to produce density maps with precise pedestrian locations, which serve as ground truth for network training. Following this, a lightweight feature extraction module (LFEM) is designed to reduce computational complexity and improve the network's overall efficiency. Lastly, a refined temporal inference module (RTIM) is developed to capture video-level semantic features, ultimately delivering accurate crowd counting results.

## 3.1 Density map generation module

Suppose the position of a person's head annotation is $x_i$, which can be represented by a shock pulse function $\delta(x-x_i)$. If there are $N$ head annotations in a crowd image, it can be represented by the following Formula 1:

$$H(x) = \sum_{i=1}^{N} \delta(x - x_i) \tag{1}$$

After annotating the crowd image, by performing convolution with a two-dimensional Gaussian kernel function $G_\sigma$, the corresponding crowd density map $F(x)$ of the image can be represented by the following Formula 2:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \cdot G_\sigma(x) \tag{2}$$

Due to the "size varies with distance" problem of head scales in the image scene, which results in significant differences in head sizes at different positions, Zhang et al. [11] proposed using a geometric adaptive Gaussian kernel $G_{\sigma_i}$ instead of a fixed-size two-dimensional Gaussian kernel function $G_\sigma$ to generate the crowd density map. In crowded scenes, the size of a head is often related to the distance between it and the centers of adjacent heads. Therefore, in such scenes, the standard deviation $\sigma_i$ of the geometric adaptive Gaussian kernel can be determined by the average distance $\overline{d_i}$ between a given head position $x_i$ and its neighboring $k$ heads. The generated crowd density map $F(x)$ is defined as following Formula 3:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \cdot G_{\sigma_i}(x) \tag{3}$$

Here, $\sigma_i = \beta \bullet \overline{d_i}$ and $\beta$ represent weight coefficients. Zhang et al. [11] demonstrated through extensive experiments that the results are optimal when $\beta = 0.3$ is used.

In the crowd density maps using the two types of Gaussian kernel functions described above, the spatial distribution information is represented by a series of blurred Gaussian spots, which cannot provide the precise locations of each person. This limits subsequent crowd analysis and practical applications. Therefore, we introduce the focal inverse distance transform (FIDT) to generate crowd density maps with accurate pedestrian locations [18]. Next, we first introduce the Euclidean Distance Transform mapping, which generates density map annotations by calculating the Euclidean distance between each pixel and its nearest annotation point. The Formula 4 is defined as follows:

$$D(x,y) = \min_{(x',y') \in S} \sqrt{(x-x')^2 + (y-y')^2} \tag{4}$$

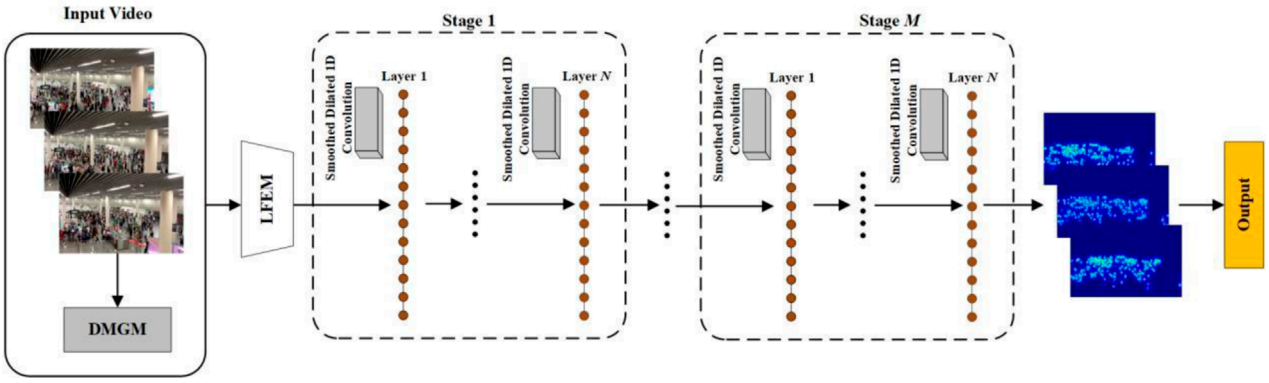



FIGURE 2
Overall structure of LMSTIN.

Here, $S$ represents the set of all head annotations, and $D(x,y)$ denotes the Euclidean distance between the head annotation position $(x,y)$ and the nearest head annotation position $(x',y')$. Due to the significant variation in distances between different heads, directly regressing the crowd density map can result in it approaching zero overall. To address this issue, the Inverse Distance Transform (IDT) can be applied to smooth out the distance variation. The Formula 5 is defined as follows:

$$I'(x,y) = \frac{1}{D(x,y) + C} \tag{5}$$

Here, $I'(x,y)$ represents the density map generated using IDT, and $C$ is a constant. To prevent the denominator from being zero, $C = 1$ is usually set. However, while the pixel values generated by IDT decay rapidly at locations far from the head annotation centers, the decay in the background is not sufficiently pronounced. Building upon this, FIDT is further proposed to make the decay near the heads slower while accelerating the decay to zero at farther locations. The Formula 6 is defined as follows:

$$I(x,y) = \frac{1}{D(x,y)^{(\alpha \cdot D(x,y)+\beta)} + C} \tag{6}$$

Here, $I(x,y)$ represents the density map generated using FIDT, and $\alpha$ and $\beta$ are set to 0.02 and 0.75, respectively.

## 3.2 Lightweight feature extraction module

The VGG-16 network, due to its excellent performance in image feature extraction, has been favored by many researchers in the field of crowd counting [19, 20]. This network consists of 13 convolutional layers with 3 × 3 kernels, 5 pooling layers with 2 × 2 kernels, and 3 fully connected layers. When applied to different tasks, the fully connected layers are usually removed, retaining only the convolutional and pooling layers to extract features from crowd images. Unlike ResNet, VGG-16 has a relatively moderate number of network layers and consumes fewer computational resources, which allows it to improve convergence speed while ensuring effective feature extraction. Nevertheless, VGG-16 still does not meet the high real-time requirements of video crowd counting tasks effectively.

Therefore, this section designs a Lightweight Feature Extraction Module (LFEM) that replaces traditional convolutions with depthwise separable convolutions to reduce network parameters, thus improving operational efficiency while achieving feature extraction results comparable to VGG-16. Depthwise separable convolution, proposed by Chollet et al. [21], is an efficient convolution operation that consists of two main steps: Depthwise Convolution and Pointwise Convolution, as shown in Figure 3. Specifically, Depthwise Convolution performs convolution operations across channels, where each channel has its own kernel, and the kernel size is the same as the traditional convolution kernel being replaced. Thus, the number of input and output channels remains consistent throughout the process. Pointwise Convolution, composed of 1 × 1 kernels, is used to weight the output features from the previous step and adjust the number of output feature channels. The number of kernels depends on the required number of output feature channels, so this process does not change the feature map size. Figure 3 illustrates the specific operation process of depthwise separable convolution. Finally, an additional fully connected layer is added to LFEM to generate a feature vector that meets the input dimensions of the next module. Experimental results indicate that, despite having significantly fewer parameters than VGG-16, LFEM can still achieve results comparable to VGG-16. This provides a solid foundation for achieving real-time performance with our method.

## 3.3 Refined temporal inference module

Modeling spatio-temporal information in video sequences has shown good performance in addressing problems such as person occlusion, background interference, and scale variation in crowd counting problems. To address these problems, we construct a refined temporal inference module (RTIM), which includes multiple stages of temporal inference modules. The output of the previous stage module serves as the input for the next stage module. Each stage's temporal inference module is composed of multiple smooth dilated 1D convolutions stacked together, with a loss layer at the end of each stage to adjust the output features. The final stage outputs the counting results. Since smooth

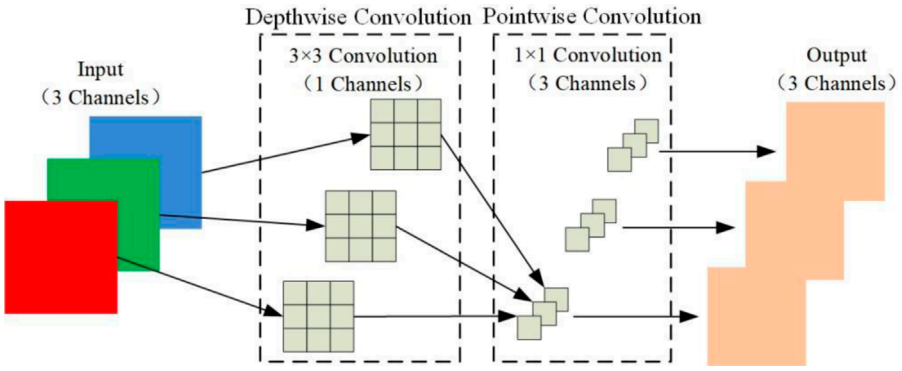



FIGURE 3
The operation process of depthwise separable convolution.

dilated 1D convolution can learn temporal information with a larger receptive field using fewer parameters [22], RTIM can maintain a low computational complexity while focusing on useful temporal information to achieve efficient and reliable temporal video modeling. The following will provide a detailed description of smooth dilated 1D convolution and the loss function.

### 3.3.1 Smooth dilated 1D convolution

Dilated convolution can effectively expand the receptive field of the filter without increasing the number of parameters and computational load, allowing it to process information over a larger area. In recent years, dilated convolution has gained widespread attention in the field of deep learning. However, it also has some drawbacks, such as the loss of local spatial information, as noted by Chen et al. [22]. Additionally, there is no dependency between input units or output units in dilated convolution, leading to ineffective acquisition of contextual information during network training [23]. For fine recognition tasks such as image segmentation and crowd counting, dilated convolution can result in the loss of local spatial information and lack of contextual information during training, severely impacting the final recognition results.Since RTIM mainly consists of a set of dilated 1D convolution layers, it also suffers from problems of local temporal information loss and lack of relevance in long-range temporal information. To address this, we introduce smooth dilated 1D convolution. Next, we will briefly introduce dilated 1D convolution and then provide a detailed description of smooth dilated 1D convolution.

For a dilated 1D convolution with a filter of size $k$ and dilation rate $w$, the output $Z$ at position $i$ is defined as following Formula 7:

$$Z[i] = \sum_{s=1}^{k} f[i + r \times s] w[i] \tag{7}$$

Here, $f$ represents the one-dimensional input, and $r$ represents the dilation rate. When $r = 1$, the dilated 1D convolution reduces to a standard 1D convolution. To intuitively understand dilated 1D convolution, it can be viewed as inserting $r - 1$ zeros between two adjacent weights of $w$. Therefore, its receptive field becomes $r \times (k - 1) + 1$.

To address the issues related to dilated convolutions, we propose a smooth dilated 1D convolution method. This approach uses "separable" and "shared" operations to smooth the dilated 1D convolution before applying the dilated 1D convolution operation. This enables the network to enhance the dependencies between local temporal features in advance, allowing it to capture a broader range of temporal context without increasing computational complexity, effectively reducing the loss of local temporal information. "Separable" refers to the separable convolution mentioned in existing literature [21], while "shared" means that the convolution weights are shared across all channels [23]. Specifically, a separable and shared convolution with a kernel size of $(2r - 1)$ is inserted before the dilated 1D convolution to capture the temporal dependencies between feature maps generated by periodic subsampling. During the smoothing operation (including "separable" and "shared"), there is only one constant parameter that is independent of the number of channels, with a size of $(2r - 1)$. Therefore, the additional computational cost is negligible. Figure 4 shows a schematic of a smooth dilated 1D convolution. As illustrated in the figure, it depicts a smooth dilated 1D convolution with a kernel size of 3 and a dilation rate of 2. The gray circles represent the feature maps after the smoothing operation, while the brown circles represent the original feature maps. Smooth dilated 1D convolution increases the dependencies between input units by adding separable and shared convolutions before the dilated 1D convolution. In short, when using smooth dilated 1D convolution, the features at non-zero positions can incorporate local temporal information from their adjacent zero-value positions. This effectively mitigates the loss of local temporal information and enhances long-range temporal dependencies.

### 3.3.2 Loss function

In crowd counting algorithms based on density maps, the Euclidean distance (denoted as $L_E$) is primarily used to measure the error between the actual and predicted crowd counts. The Formula 8 is defined as follows:

$$L_E = \frac{1}{N} \sum_{i=1}^{N} \left( C_i^p - C_i^{gt} \right)^2 \tag{8}$$

Here, $N$ represents the number of frames in the video, $C_i^p$ denotes the estimated count for the image in frame $i$, and $C_i^{gt}$ denotes the actual count for the image in frame $i$. Although $L_E$

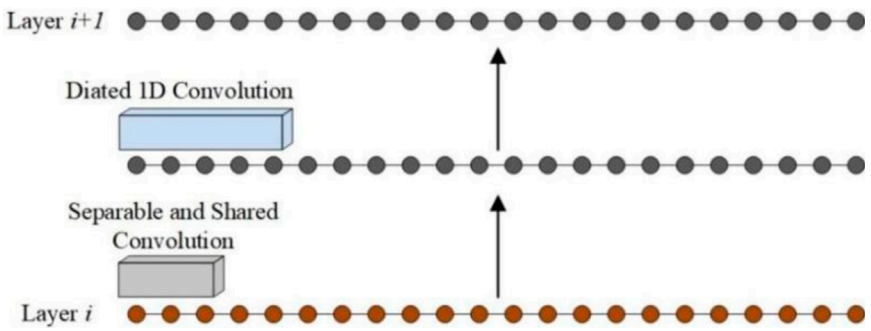



FIGURE 4
Schematic of smooth dilated 1D convolution.

loss has performed well in image crowd counting tasks, it does not account for spatio-temporal consistency in video sequences. To further improve the accuracy of video crowd counting, this section introduces a smoothing loss (denoted as $L_S$) by incorporating the similarity between video frames to reduce prediction errors between consecutive video frames. The Formulas 9–11 is defined as follows:

$$L_S = \frac{1}{N}\sum_i \tilde{\Delta}_i^2 \tag{9}$$

$$\tilde{\Delta}_i = \begin{cases} \Delta_i: & \Delta_i \le \tau \\ \tau: & \text{otherwise} \end{cases} \tag{10}$$

$$\Delta_i = \left| log C_i^p - log C_{i-1}^p \right| \tag{11}$$

Here, $\tau$ represents the hyperparameter $L_S$. Combining the above loss functions, the final form of the loss function is as following Formula 12:

$$L = L_E + \lambda L_S \tag{12}$$

Here, $\lambda$ represents the hyperparameter that adjusts the weight of $L_S$. The values of all hyperparameters will be provided in the subsequent experimental section.

# 4 Experimental setup

## 4.1 Implementation details

The experiments are implemented using PyTorch for LMSTIN. The RTIM consists of four stages, each with 10 smooth dilated 1D convolutional layers, where the dilation rate of each layer is twice that of the previous layer. After each convolutional layer, a dropout with a rate of 0.5 is applied, with a kernel size of 3 and 64 convolutional filters. Additionally, the loss function of LMSTIN is a combination of Euclidean distance loss and smoothing loss, with the parameters set to $\tau = 10$ and $\lambda = 0.15$. In all experiments, Adam is used to optimize the network parameters, with a learning rate of 0.0005 and no weight decay.

## 4.2 Datasets

In this paper, we evaluate the performance of the proposed LMSTIN on three public video crowd counting datasets: Mall [24], UCSD [25], and WorldExpo'10 [10]. The Mall dataset was collected using surveillance cameras installed in a shopping mall. It consists of 2000 frames of video with a resolution of 320 × 240 pixels per frame, and a total of 62,325 pedestrians are labeled. The number of people per frame ranges from a minimum of 11 to a maximum of 53, with an average of approximately 31 people per frame. The Mall dataset features high crowd density and diverse scenes, and it is divided into a training set and a test set, with the first 800 frames used for training and the remaining 1200 frames used for testing.The UCSD dataset was collected using cameras installed in a pedestrian-only corridor at the University of California, San Diego. The original videos were collected at a resolution of 740 × 480 and a frame rate of 30 FPS, then downsampled to 238 × 158 and 10 FPS. The UCSD dataset contains 2000 frames with a total of 49,885 labeled pedestrians. To exclude unnecessary objects (such as trees and cars), an interest region is defined within which annotations are made manually every 5 frames, with linear interpolation used for the remaining frames. The UCSD dataset is collected from a fixed position, so the scene perspective remains unchanged throughout the video.The WorldExpo'10 dataset is a large-scale cross-scene crowd counting dataset. It was collected from the 2010 Shanghai Expo, including 1132 video sequences with manual annotations captured by 108 surveillance cameras. The dataset consists of 3920 frames with a resolution of 576 × 720 pixels, and a total of 199,923 people are labeled, with an average of 50 people per frame.

## 4.3 Evaluation metrics

The experiments use two evaluation metrics, namely, Mean Absolute Error (MAE) and Mean Squared Error (MSE), to assess the accuracy and robustness of the method. The specific formulas are as following Formulas 13, 14:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left| C_i^p - C_i^{gt} \right| \tag{13}$$

TABLE 1 Comparison of our method with existing methods on the mall dataset.

| Methods | MAE | MSE |
|---|---|---|
| Gaussian Process Regression [25] | 3.72 | 20.10 |
| Ridge Regression [24] | 3.59 | 19.00 |
| Kernel Ridge Regression [26] | 3.51 | 18.10 |
| Cumulative Attribute Regression [27] | 3.43 | 17.70 |
| Count Forest [28] | 2.50 | 10.00 |
| ConvLSTM [1] | 2.24 | 8.50 |
| Bidirectional ConvLSTM [1] | 2.10 | 7.60 |
| LSTN [2] | 2.00 | 2.50 |
| MLSTN [6] | 1.80 | 2.42 |
| E3D [4] | 1.64 | 2.13 |
| Monet [29] | 1.54 | 2.02 |
| STDNet [5] | 1.47 | 1.88 |
| Ours | **1.40** | **1.76** |

TABLE 2 Comparison of our method with existing methods on the UCSD dataset.

| Methods | MAE | MSE |
|---|---|---|
| Ridge Regression [24] | 2.25 | 7.82 |
| Gaussian Process Regression [25] | 2.24 | 7.97 |
| Kernel Ridge Regression [26] | 2.16 | 7.45 |
| Cumulative Attribute Regression [27] | 2.07 | 6.86 |
| Switch-CNN [30] | 1.62 | 2.10 |
| Cross-Scene [10] | 1.60 | 3.31 |
| FCN-rLSTM [31] | 1.54 | 3.02 |
| ConvLSTM [1] | 1.30 | 1.79 |
| Monet [29] | 1.17 | 1.45 |
| Bidirectional ConvLSTM [1] | 1.13 | 1.43 |
| LSTN [2] | 1.07 | 1.39 |
| MLSTN [6] | 1.02 | 1.32 |
| E3D [4] | 0.93 | 1.17 |
| STDNet [5] | 0.76 | 1.01 |
| Ours | **0.71** | **0.94** |

Bold font indicates the best value of the evaluation Metrics.

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i^p - C_i^{gt}\right)^2} \tag{14}$$

Here, $N$ represents the number of frames in the video, and $C_i^p$ and $C_i^{gt}$ denote the estimated count and the actual count for the image in frame $i$, respectively. MAE measures the accuracy of the counting method, while MSE evaluates the robustness of the counting method. The smaller the values of MAE and MSE, the better the accuracy and robustness of the method, and thus, the better its performance.

# 5 Experimental results and analysis

## 5.1 Quantitative and qualitative analysis

We compared the crowd counting results of our proposed method with several state-of-the-art video crowd counting methods on the Mall, UCSD, and WorldExpo'10 datasets. The comparison results are shown in Tables 1–3.

Comparing with 12 advanced video crowd counting methods, our proposed LMSTIN achieved the best results across all metrics, as detailed in Table 1. From Table 1, it can be observed that LMSTIN shows a further improvement over the current state-of-the-art method (STDNet), reducing the MAE by 0.07 and the MSE by 0.12. Additionally, the Mall dataset presents more complex scenarios compared to the UCSD dataset, such as higher levels of perspective distortion and occlusion, which can lead to inaccuracies or imprecisions in annotations. In this context, LMSTIN addresses this problem by modeling spatio-temporal consistency between video frames. Experimental results demonstrate that LMSTIN effectively models temporal dependencies between video frames, thereby extracting more robust spatio-temporal features to enhance the network's capability for crowd counting tasks.

Table 2 presents a comparison of LMSTIN with 14 state-of-the-art video crowd counting methods. The experimental results show that LMSTIN outperforms all previous methods in both MAE and MSE metrics, achieving reductions of 0.05 in MAE and 0.07 in MSE compared to STDNet.Notably, the improvements on the UCSD dataset have two important implications. First, with a frame rate of 10 FPS, the UCSD dataset allows the network to learn multi-scale temporal features due to the high correlation between consecutive frames. For instance, in a video segment with 20 frames, if a person appears continuously from frame 1 to frame 20, LMSTIN can extract both short-term information (e.g., from frame 1 to frame 2) and long-term information (e.g., from frame 1 to frame 20) from the video frames. Second, since individuals typically move at varying speeds, multi-scale temporal information helps account for people moving at different velocities, which is beneficial for density map estimation in crowded scenes. The experimental results indicate that effectively modeling both short-term and long-term temporal information provides robust performance against crowd occlusion and scale variations in complex environments, leading to improved crowd counting results.

Table 3 summarizes the experimental results of LMSTIN compared with 9 state-of-the-art video crowd counting methods.

TABLE 3 Comparison of our method with existing methods on the WorldExpo'10 dataset.

| Methods | S1 | S2 | S3 | S4 | S5 | Avg |
|---|---|---|---|---|---|---|
| Cross-Scene [10] | 9.8 | 14.1 | 14.3 | 22.2 | 3.7 | 12.9 |
| ConvLSTM-nt [1] | 8.6 | 16.9 | 14.6 | 15.4 | 4.0 | 11.9 |
| Switch-CNN [30] | 4.4 | 15.7 | 10 | 11 | 5.9 | 9.4 |
| ConvLSTM [1] | 7.1 | 15.2 | 15.2 | 13.9 | 3.5 | 10.9 |
| Bidirectional ConvLSTM [1] | 6.8 | 14.5 | 14.9 | 13.5 | 3.1 | 10.6 |
| ST-CNN [32] | 5.2 | 16.5 | 9.9 | 8.4 | 6.2 | 9.3 |
| E3D [4] | 2.8 | 12.5 | 12.9 | 10.2 | 3.2 | 8.3 |
| TAN [33] | 2.8 | 18.1 | 9.6 | 7.5 | 3.6 | 8.3 |
| STDNet [5] | 1.8 | **12.8** | 10.3 | 7.9 | **2.5** | 7.1 |
| Ours | **1.6** | 14.3 | **8.2** | **7.0** | 2.8 | **6.8** |

Bold font indicates the best value of the evaluation Metrics.

In this experiment, 16 consecutive frames were used as input, and MAE and average MAE (Avg) across 5 scenes (S1, S2, S3, S4, S5) were used as evaluation metrics. The results show that, compared to the current best method STDNet, LMSTIN has achieved an overall improvement in accuracy, reducing the average MAE by 0.3. However, its performance varies across different scenes: it decreased by 0.2, 2.1, and 0.9 in scenes S1, S3, and S4, respectively, but increased by 1.5 and 0.3 in scenes S2 and S5. This discrepancy is because the temporal correlation between consecutive frames in scenes S2 and S5 is not strong, and these scenes are relatively sparse, which conflicts with our design objectives. Nevertheless, LMSTIN still achieved the best accuracy in 3 out of 5 scenes and provided the lowest average MAE (Avg). This indicates that LMSTIN not only effectively models both short-term and long-term video temporal information but also demonstrates good robustness across datasets with varying scales and scene differences.

In order to facilitate observation and comparative analysis, the final crowd density maps generated by LMSTIN and STDNet are visualized respectively, because this method is one of the most advanced methods in the field of video crowd counting. The visualization results are shown in Figure 5. In Figure 5, the first row is the visualization result of the Mall dataset, the second row is the visualization result of the UCSD dataset, and the third row is the visualization result of the WorldExpo'10 dataset. The first column is the input original image, the second column is the FIDT real density map, and the third and fourth columns represent the corresponding output density maps of STDNet and the method in this chapter, respectively. The numbers in the figure represent the real annotation (GT) and the predicted number of people (Pred). As can be seen from Figure 5, the density map generated by the method in this chapter is closer to the real density map than the density map generated by STDNet, so the counting results and pedestrian locations are also more accurate. The visualization results intuitively demonstrate the effectiveness and robustness of the method in this chapter on the video crowd counting task, and the output crowd density map can provide accurate pedestrian location information, which provides the necessary prerequisite for subsequent crowd analysis tasks.

## 5.2 Structural analysis and efficiency comparison

To validate the effectiveness of each module in LMSTIN, we first analyze the impact of different structures on video crowd counting results by examining LFEM and RTIM. Then, we compare LMSTIN with current state-of-the-art video crowd counting methods from multiple aspects to demonstrate LMSTIN's real-time performance and effectiveness.

First, we evaluate the performance of LFEM. The VGG-16 network, known for its excellent feature extraction capabilities, has become a mainstream feature extraction method in the field of crowd counting. Specifically, the VGG-16 network consists of 16 layers, including 13 convolutional layers with $3 \times 3$ kernels, 5 pooling layers with $2 \times 2$ kernels, and 3 fully connected layers. In crowd counting tasks, the fully connected layers of VGG-16 are typically discarded, retaining only the convolutional and pooling layers to extract features from crowd images. Our proposed LFEM is an improvement based on the VGG-16 network, aiming to reduce the computational load of the network while maintaining its feature extraction capabilities, thus enhancing the overall operational efficiency of the network. Therefore, in the ablation experiments evaluating LFEM's performance, in addition to using MAE, MSE, and Avg as evaluation metrics, we also introduce the number of module parameters (Params) as an important indicator of computational complexity. Table 4 presents the experimental results of LFEM and VGG-16 on the three datasets.

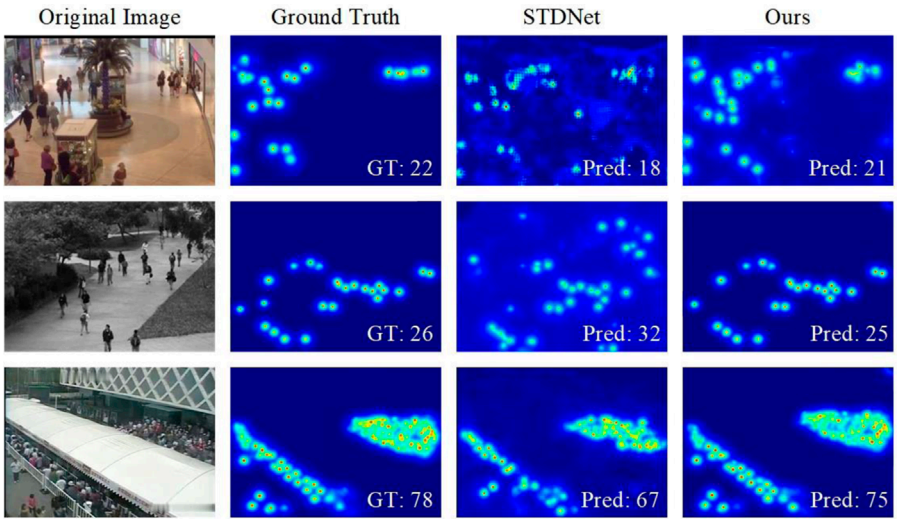



FIGURE 5
Qualitative results.

TABLE 4 Comparison of Experimental Results between LFEM and VGG-16 on three Datasets.

| Methods | Datasets | MAE | MSE | Avg | Params |
|---|---|---|---|---|---|
| VGG-16 | Mall | 1.38 | 1.72 | — | 0.16M |
| | UCSD | 0.71 | 0.93 | — | |
| | WorldExpo'10 | — | — | 6.6 | |
| LFEM | Mall | 1.40 | 1.76 | — | 0.03M |
| | UCSD | 0.71 | 0.93 | — | |
| | WorldExpo'10 | — | — | 6.8 | |

To visually compare the performance differences between LFEM and VGG-16, the experiment involved replacing only the feature extraction module in the entire method while keeping the other modules unchanged. From Table 4, it is evident that LFEM performs similarly to VGG-16 across all datasets, achieving the same accuracy on the UCSD dataset. However, LFEM's parameter count is only about one-fifth of that of VGG-16. The experimental results demonstrate that LFEM is an effective feature extraction module for video crowd counting tasks, significantly reducing the network's computational complexity and thereby enhancing the overall efficiency of the method.

Next, we evaluate the performance of RTIM. In crowd counting tasks, the current methods for modeling spatio-temporal relationships between video frames mainly use LSTM, Bi-directional LSTM (BI-LSTM), and 3DCNN as the foundational frameworks. To intuitively compare the performance differences between RTIM and other temporal modeling networks, we replace only the temporal inference part in the entire method, keeping the other modules unchanged. Since the ablation experiments yield consistent conclusions across the three datasets, we present the results using the Mall dataset as an example. The results are shown in Table 5.

Table 5 lists the MAE, MSE, and Params for different temporal modeling networks tested on the Mall dataset. From Table 5, it is evident that RTIM significantly improves accuracy compared to LSTM, BI-LSTM, and 3DCNN, while also substantially reducing the network parameter count. RTIM achieves the best results in both MAE and MSE and has less than one-seventh of the network parameters compared to 3DCNN. The experimental results demonstrate that RTIM can effectively model the temporal relationships between video frames with minimal parameters, thus further enhancing the overall efficiency of the method. This is critically important for the practical application of video crowd counting methods.

Finally, the overall operating efficiency of LMSTIN is evaluated. Taking the Mall dataset as an example, the differences in operating efficiency of the method in this chapter are illustrated by comparing it with four state-of-the-art video crowd counting methods, STDNet, Monet, E3D, and MLSTN. Table 6 lists the parameter amount (Params), computation amount (FLOPs), and training time

TABLE 5 Comparison of RTIM with other temporal modeling networks on the Mall Dataset.

| Methods | MAE | MSE | Params |
|---|---|---|---|
| LSTM [34] | 2.25 | 6.50 | 2.31M |
| BI-LSTM [35] | 2.02 | 4.65 | 4.65M |
| 3DCNN [36] | 1.68 | 2.20 | 5.70M |
| RTIM | **1.40** | **1.76** | **0.82M** |

Bold font indicates the best value of the evaluation Metrics.

TABLE 6 Comparison of parameters, computation and training time of different methods on the Mall dataset.

| Methods | Params | FLOPs | Training time |
|---|---|---|---|
| MLSTN [6] | 12.25M | 56.50M | 53Mins |
| Monet [29] | 11.58M | 41.65M | 47Mins |
| E3D [4] | 6.42M | 23.20M | 30Mins |
| STDNet [5] | 2.80M | 5.76M | 18Mins |
| Ours | **0.85M** | **2.74M** | **12Mins** |

Bold font indicates the best value of the evaluation Metrics.

(Training Time) of different networks. As can be seen from Table 6, LMSTIN is significantly more efficient than networks such as STDNet, Monet, E3D, and MLSTN. For example, in terms of computation amount, the value of STDNet is about 2.5 times that of the method in this chapter, the value of E3D is about 11 times that of E3D, the value of Monet is about 20 times that of E3D, and the value of MLSTN is about 23 times that of E3D. In terms of parameter amount, the value of STDNet is about 3 times that of the method in this chapter, the value of E3D is about 8 times that of E3D, the value of Monet is about 13 times that of E3D, and the value of MLSTN is about 14 times that of E3D. The training time in Table 3.6 is the running time for training the Mall dataset for 50 cycles (Epoch) on a single GTX TitanXp GPU. It can be seen that the training time of this chapter's method is shorter than that of all other networks. The experimental results show that this chapter's method is significantly better than the existing methods in terms of network parameters, computational complexity and running time, and the overall network operation efficiency has been significantly improved compared with other methods. It is worth noting that for videos with a resolution of 320 × 240 pixels, this chapter's method only occupies less than 500 MB of GPU memory on the Nvidia TitanXp GPU to achieve a detection speed of 120FPS, and also achieves a real-time crowd counting speed of 25FPS on the daily home Intel Core i5-8400 CPU.

Overall, extensive experiments demonstrate that each module within LMSTIN, performs exceptionally well, significantly surpassing existing advanced methods in both speed and accuracy. This advancement has substantial implications for the practical application of crowd counting methods in real-world monitoring scenarios.

## 6 Conclusion

We propose a lightweight multi-stage temporal inference network for video crowd counting, named LMSTIN. Specifically, LMSTIN first utilizes the focal inverse distance transformation to convert input video frames into crowd density maps with accurate pedestrian locations, which serve as the ground truth labels for network training. Secondly, we design a lightweight feature extraction module to reduce the computational load of the model, enhancing overall efficiency while maintaining effective spatial feature extraction. Finally, we build a multi-stage temporal inference module with minimal parameters that performs well, focusing on modeling temporal relationships to efficiently extract spatio-temporal information from video frames. Experimental results demonstrate that our method achieves excellent performance across various datasets and is capable of real-time crowd counting at 25 frames per second on an Intel Core i5-8400 CPU. LMSTIN have great potential for future development, especially in handling more complex video scenes, different crowd movement patterns, and integrating other functionalities. By combining with features like behavior recognition, they can achieve comprehensive monitoring and analysis of crowd behavior, providing stronger technical support for public safety, traffic management, and business decision-making.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

## Author contributions

WG: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Writing–original draft, Writing–review and editing. RF: Data curation, Formal Analysis, Investigation, Methodology, Supervision, Writing–review and editing. XS: Investigation, Validation, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work is funded by National Social Science Foundation Project (Project Title: Research on the Generation Mechanism and Public Governance of Online Disputes in Public Events, Grant No. 20BXW109) and the 2020 general project of philosophy and social sciences research in colleges and universities of Jiangsu Province (Project Title: Research and Evaluation of Learning Performance based on MOOC platform, Grant No. 2020SJA1173).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

1. Xiong F, Shi X, Yeung DY. Spatiotemporal modeling for crowd counting in videos. *Proc IEEE Int Conf Computer Vis* (2017) 5151–9. doi:10.1109/ICCV.2017.551

2. Fang Y, Zhan B, Cai W, et al. Locality-constrained spatial transformer network for video crowd counting. In: *2019 IEEE international conference on multimedia and Expo (ICME)*. IEEE (2019) p. 814–9.

3. Wu X, Xu B, Zheng Y, et al. *Video crowd counting via dynamic temporal modeling* (2019) p. 19. arXiv:1907.02198.

4. Zou Z, Shao H, Qu X, et al. Enhanced 3D convolutional networks for crowd counting. arXiv:1908.04121 (2019). doi:10.48550/arXiv.1908.04121

5. Ma YJ, Shuai HH, Cheng WH. Spatiotemporal dilated convolution with uncertain matching for video-based crowd estimation. *IEEE Trans Multimedia* (2021) 24:261–73. doi:10.1109/tmm.2021.3050059

6. Fang Y, Gao S, Li J, Luo W, He L, Hu B. Multi-level feature fusion based locality-constrained spatial transformer network for video crowd counting. *Neurocomputing* (2020) 392:98–107. doi:10.1016/j.neucom.2020.01.087

7. Fu M, Xu P, Li X, Liu Q, Ye M, Zhu C. Fast crowd density estimation with convolutional neural networks. *Eng Appl Artif Intelligence* (2015) 43:81–8. doi:10.1016/j.engappai.2015.04.006

8. Wang C, Zhang H, Yang L, et al. *Deep people counting in extremely dense crowds Proceedings of the 23rd ACM International Conference on Multimedia* (2015) p. 1299–1302.

9. Krizhevsky A, Sutskever I, Hinton GE. Imagine classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* (2012) 25. doi:10.1145/3065386

10. Zhang C, Li H, Wang X, et al. Cross-scene crowd counting via deep convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) p. 833–41. doi:10.1109/CVPR.2015.7298684

11. Zhang Y, Zhou D, Chen S, et al. Single-image crowd counting via multi-column convolutional neural network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) p. 589–97.

12. Zhang A, Shen J, Xiao Z, et al. Relational attention network for crowd counting. *Proc IEEE/CVF Int Conf Computer Vis* (2019) 6788–97. doi:10.1109/ICCV.2019.00689

13. Hu R, Tang ZR, Wu EQ, Mo Q, Yang R, Li J. RDC-SAL: refine distance compensating with quantum scale-aware learning for crowd counting and localization. *Appl Intelligence* (2022) 52(12):14336–48. doi:10.1007/s10489-022-03238-4

14. Liu N, Long Y, Zou C, et al. Adcrowdnet: an attention-injective deformable convolutional network for crowd understanding. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019) p. 3225–34.

15. Liang D, Chen X, Xu W, Zhou Y, Bai X. TransCrowd: weakly-supervised crowd counting with transformers. *Sci China Inf Sci* (2022) 65(6):160104–14. doi:10.1007/s11432-021-3445-y

16. Li B, Zhang Y, Xu H, Yin B. CCST: crowd counting with swin transformer. *Vis Computer* (2022) 39:2671–82. doi:10.1007/s00371-022-02485-3

17. Bai H, He H, Peng Z, et al. *CounTr: an end-to-end transformer approach for crowd counting and density estimation European conference on computer vision*. Cham: Springer (2023) p. 207–222.

18. Liang D, Xu W, Zhu Y, et al. Focal inverse distance transform maps for crowd localization and counting in dense crowd. *arXiv preprint arXiv:2102.07925* (2021). doi:10.1109/CVPR.2016.70

19. Cao X, Wang Z, Zhao Y, et al. Scale aggregation network for accurate and efficient crowd counting. In: *Proceedings of the European conference on computer vision*. ECCV (2018) p. 734–50.

20. Hossain MA, Cannons K, Jang D, et al. Video-based crowd counting using a multi-scale optical flow pyramid network. In: *Proceedings of the asian conference on computer vision* (2020).

21. Chollet F. Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) p. 1251–8.

22. Chen W, Chai Y, Qi M, Sun H, Pu Q, Kong J, et al. Bottom-up improved multistage temporal convolutional network for action segmentation. *Appl Intelligence* (2022) 52(12):14053–69. doi:10.1007/s10489-022-03382-x

23. Wang Z, Ji S. Smoothed dilated convolutions for improved dense prediction. *Data Mining Knowledge Discov* (2021) 35(4):1470–96. doi:10.1007/s10618-021-00765-5

24. Chen K, Loy CC, Gong S, et al. Feature mining for localised crowd counting. *Bmvc* (2012) 1(2):3.

25. Chan AB, Liang ZSJ, Vasconcelos N. Privacy preserving crowd monitoring: counting people without people models or tracking. In: *2008 IEEE conference on computer vision and pattern recognition*. IEEE (2008) p. 1–7. doi:10.1109/CVPR.2008.4587569

26. An S, Liu W, Venkatesh S. Face recognition using kernel ridge regression. In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE (2007) p. 1–7. doi:10.1109/CVPR.2007.383105

27. Chen K, Gong S, Xiang T, et al. Cumulative attribute space for age and crowd density estimation. *Proc IEEE Conf Computer Vis Pattern Recognition* (2013) 2467–74.

28. Pham VQ, Kozakaya T, Yamaguchi O, et al. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. *Proc IEEE Int Conf Computer Vis* (2015) 3253–61. doi:10.1109/ICCV.2015.372

29. Bai H, Chan SHG. Motion-guided non-local spatial-temporal network for video crowd counting. *arXiv preprint arXiv:2104.13946* (2021).

30. Babu Sam D, Surya S, Venkatesh Babu R. Switching convolutional neural network for crowd counting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) p. 5744–52.

31. Zhang S, Wu G, Costeira JP, et al. Fcn-rlstm: deep spatio-temporal neural networks for vehicle counting in city cameras. *Proc IEEE Int Conf Computer Vis* (2017) 3667–76.

32. Miao Y, Han J, Gao Y, Zhang B. ST-CNN: spatial-temporal convolutional neural network for crowd counting in videos. *Pattern Recognition Lett* (2019) 125:113–8. doi:10.1016/j.patrec.2019.04.012

33. Wu X, Xu B, Zheng Y, Ye H, Yang J, He L. Fast video crowd counting with a temporal aware network. *Neurocomputing* (2020) 403:13–20. doi:10.1016/j.neucom.2020.04.071

34. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* (1997) 9(8):1735–80. doi:10.1162/neco.1997.9.8.1735

35. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* (2005) 18(5-6):602–10. doi:10.1016/j.neunet.2005.06.042

36. Ji S, Xu W, Yang M, Yu K. 3D convolutional neural networks for human action recognition. *IEEE Trans Pattern Anal Machine Intelligence* (2012) 35(1):221–31. doi:10.1109/tpami.2012.59

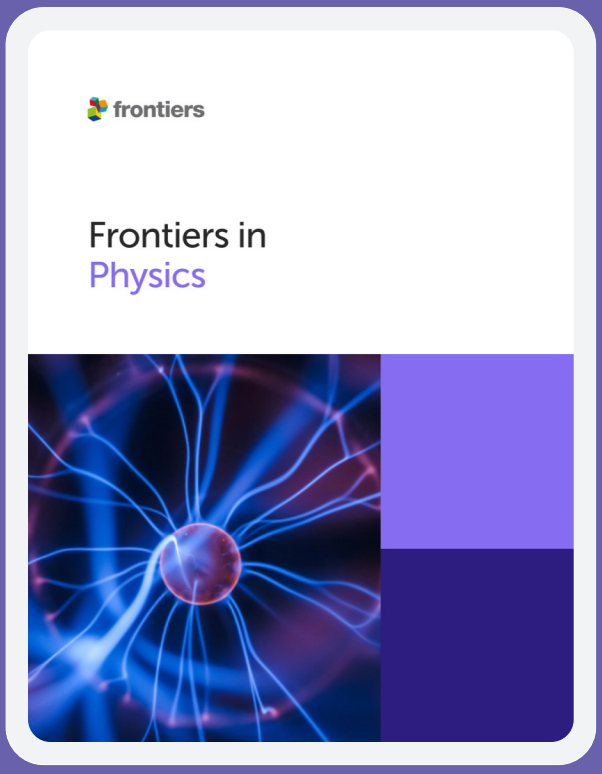
# Frontiers in Physics

**Investigates complex questions in physics to understand the nature of the physical world**

Addresses the biggest questions in physics, from macro to micro, and from theoretical to experimental and applied physics.

## Discover the latest Research Topics

See more →




**Frontiers**

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

**Contact us**

+41 (0)21 510 17 00
frontiersin.org/about/contact