



OPEN ACCESS

EDITED BY
Alex Hansen,
NTNU, Norway

REVIEWED BY
Sandeep Singh Sengar,
Cardiff Metropolitan University,
United Kingdom
Steffen Berg,
Shell Global Solutions International B.V.,
Netherlands

*CORRESPONDENCE
Muhammad Sahimi,
✉ moe@usc.edu

RECEIVED 23 December 2025
REVISED 18 December 2025
ACCEPTED 26 December 2025
PUBLISHED 17 February 2026

CITATION
Sahimi A and Sahimi M (2026) Machine
learning and digital images of porous
materials: from rock to the human brain.
Front. Phys. 13:1742403.
doi: 10.3389/fphy.2025.1742403

COPYRIGHT
© 2026 Sahimi and Sahimi. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](#). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with
these terms.

Machine learning and digital images of porous materials: from rock to the human brain

Ali Sahimi¹ and Muhammad Sahimi^{2*}

¹Grossman School of Medicine, New York University, New York, NY, United States, ²Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA, United States

Porous media and materials are ubiquitous and found everywhere. Some of them are referred to as rock-like porous media (RLPM), which include soil, concrete, asphalt, and oil and gas reservoirs. A second group consists of biological porous materials (BPMs), ranging from skin to organs such as the brain and lungs. The use of digital images of BPMs for the diagnosis and treatment of illnesses has a relatively long history, whereas their utilization in modeling various phenomena in RLPM is relatively recent. Due to the complexity of such images, along with the need to extract as much information from them as possible, the use of machine-learning (ML) approaches—in particular, neural networks (NNs)—has been increasing at a rapid pace. We describe and discuss recent progress in the applications of ML algorithms, particularly NNs, for the characterization of such images for the two classes of porous media and materials and show that, while they may seem vastly different, they actually have many similarities, and similar issues must be addressed when using and analyzing the images. As a result, the application of ML algorithms to both types of porous materials is largely similar, even though the goals may be very different.

KEYWORDS

biological porous media, digital image, machine learning, neural network, rock-like porous material

1 Introduction

Porous media and materials are ubiquitous and found everywhere, from the nanoscale at the molecular level to the largest, gigascopic scales encountered in landscapes, mountains, and rivers [1]. Almost all natural porous media and materials, and many synthetic ones, are also *disordered* or *heterogeneous* over some length scales. The type of heterogeneity depends on the length scale of observation and measurement. At small or laboratory scales, heterogeneity manifests itself through the morphology of the pore space, which consists of its geometry and topology. The geometry is represented by spatial variations in pore size, the porosity of the pore space (the pores' volume fraction), and the roughness of the internal pore surfaces. The topology describes the interconnectivity of the pores and how pore connectivity at small scales affects macroscopic properties of the pore space at large scales, along with the tortuosity τ of the pore space [2], which is a measure of the actual length of the paths for fluid flow, transport, and reaction in the pore space. Due to such complexities, the characterization of porous media and materials has always been an active area of research.

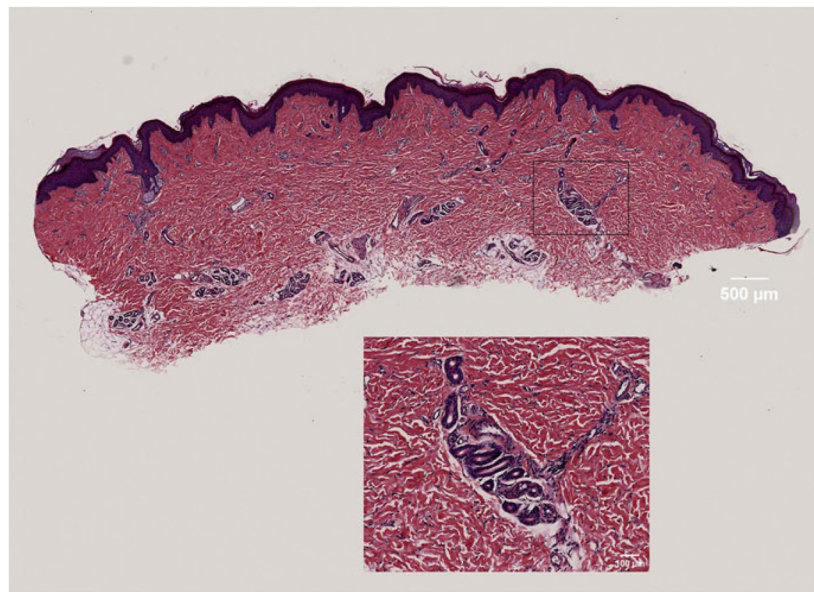


FIGURE 1

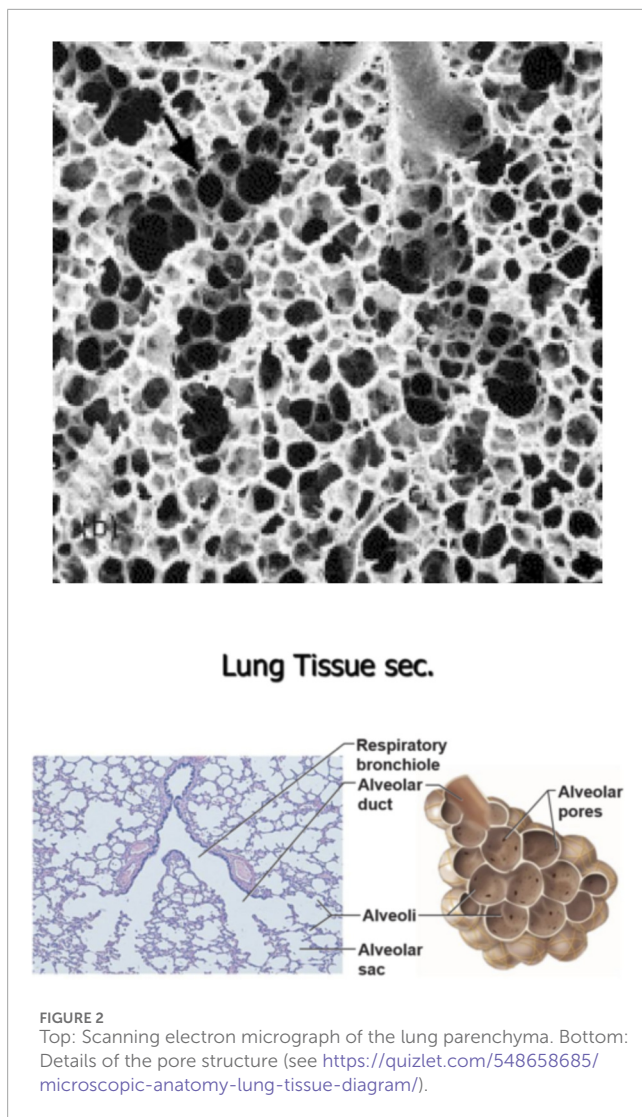
High-resolution image of pore structures in skin tissue [5]. Skin fibers are shown in pink, the white space between the fibers represents the pores, and the purple structures that are boxed and zoomed in on are the sweat glands. The porosity is approximately 0.19, while the mean pore diameter is approximately $220 \pm 34 \mu\text{m}$. A fully automatic microscope was used to scan the skin section.

Two important classes of porous materials are referred to as rock-like porous media (RLPM) and biological porous materials (BPMs). The former includes soil, sandstone, concrete, and several other types of porous media. In the latter group are skin, lung, brain, liver, bone, and many other types of biological tissues and organs. The properties of several types of RLPM have been studied, characterized, and known for decades, while there is still significant ongoing research on other types. Recognizing biological materials as porous media is a much more recent development; therefore, the characterization of their morphology is a subject of ongoing research. It is, therefore, useful to discuss some of the important properties of BPMs.

The outer layer of the epidermis of human skin is referred to as the stratum corneum, which protects the underlying tissues from infection, dehydration, chemicals, and mechanical stress. It is made up of up to twenty layers of flattened cells whose properties include mechanical shear and impact resistance, water flux and hydration regulation, and selective permeability that excludes toxins and other harmful materials. The pore structure of stratum corneum has been characterized [3, 4] by Raman scattering, as well as optical, chemical, and electron microscopy. Figure 1 shows a high-resolution image of the pore structure of human skin.

Lungs have long been viewed as porous media [6, 7] and have been modeled as such to study the various phenomena that occur within them. In particular, their pore-size distribution [8], their tortuosity, and the fractal dimension D_f of the three-dimensional (3D) pore space [9] have been measured and characterized. Measurements indicated [8] that the pore sizes vary between 5 and 100 nm, while tortuosity was $\tau \approx 1.22$, and the fractal dimension turned out to be $D_f \approx 2.34$ [9]. Figure 2 presents an image of the lung along with an explanation of the various parts of the organ.

Digitized images indicate that the microstructure of the interstitial space (IS) of the brain's extracellular space resembles an unconsolidated (granular) porous medium. It is a porous medium because it is a fluid-saturated, two-phase material. One phase is the solid matrix of the cells, while the second phase is a network of pores, or void space, which contains the fluid that makes up roughly 80 percent of the brain, with the remainder being the solid matrix, thereby endowing it with characteristic poroelastic properties. The porosity, or what neuroscientists call the *volume fraction*, has important physical functions without which the brain could not operate. These functions include (a) allowing the flow of interstitial and cerebrospinal fluids that deliver nutrients, maintaining a stable ionic environment for neuronal signals, and disposing of metabolic "waste" products, such as the protein amyloid- β ; (b) allowing diffusion of neurotransmitters and ions in the pore network of extracellular space, which plays a fundamental role for cell-to-cell communication; and (c) providing mechanical protection through the fluid that absorbs physical shocks and regulates intracranial pressure. The interstitial space is, of course, deformable, and therefore, it should be viewed as a "soft" porous medium. Being a vital organ, even small deformations can result in large changes in the properties of the pore space of the IS, which can have important implications for its biological and neurological activities. Although the brain has been modeled [10] as an unconsolidated porous medium and its various microstructural properties, ranging from porosity to tortuosity and pore-size distribution, have been determined and compared with data [10, 11], such models did not take into account the effect of deformation, whereas studies of permeability and ionic conductivity of deformable porous media in other contexts have indicated that deformation strongly influences the flow and electrical conductivity [12, 13].



Of particular interest are the types of transport processes that occur in the pore space of the IS of the brain. Naturally, diffusion is one mechanism of transport. The measurement of diffusion of small molecules in the IS pore space indicated [14, 15] that the effective diffusivity D is smaller than its value D_0 in the bulk (outside the pore space), which provided strong evidence for the IS of brain's extracellular space being a porous medium since D and D_0 are related through the relation $D = \phi D_0 / \tau$, where ϕ is the porosity and τ is the tortuosity. Moreover, since the pores of the IS are very small, anomalous diffusive behavior may be expected. For example, it is known that when the ratio of the size of the diffusing molecules and that of the pores is nonzero, diffusion is hindered [16, 17], i.e., slowed down. In addition, experience with very small nanotubes, of the size that can be found in the range of pore sizes of the IS, indicated that [18] the Stokes–Einstein relation between the viscosity and diffusivity breaks down, which could also be the case in the pore space of the IS. This was speculated on [11] a long time ago. Most natural porous media also contain dead-end pores where diffusing molecules may spend a long time, giving rise to non-Gaussian diffusion, and the pore space of the IS is no exception [19]. The

second mode of transport, convection, and its role in the transport of molecules within the IS pore space have been a controversial issue. Figure 3 presents [19, 20] a realistic model of brain extracellular space and compares it with the two-dimensional section of neuropil ultrastructure observed in electron microscopy.

With advances in the development of efficient computational methods, techniques, and instrumentation for obtaining high-resolution digitized images of porous media, attention has increasingly focused on using such images not only to characterize their morphology but also to carry out direct numerical simulations of various phenomena in their pore space. Digital rock physics combines microtomographic imaging with efficient computer simulations in order to compute the effective properties of porous media and is intended to complement laboratory studies and measurements in order to gain a deeper understanding of physical processes related to various fluid flow, diffusion, reaction, and deformation processes in such media. The approach has been used extensively for RLPM. At the same time, imaging technologies play a key role in the diagnosis of abnormalities and medical therapy. Several such technologies exist, ranging from X-ray radiography and computed tomography (CT) to magnetic resonance imaging (MRI) and spectroscopy, medical optical imaging, and ultrasonic and electrical impedance tomography. Such techniques have many applications in the diagnosis of myocardial diseases, cancer of various tissues, neurological disorders, congenital heart disease, complex bone fractures, and other serious medical conditions [21].

Medical images (MIs), as well as those for RLPM, are complex systems that contain a significant amount of information whose extraction is not straightforward. A key factor in their utility is their resolution, while a second key factor is the ability of the user to extract the maximum amount of information from them. The role of such factors has motivated the application of machine-learning (ML) algorithms, particularly neural networks (NNs), to digitized images as these algorithms and tools have distinguished themselves by their ability to learn complex patterns and extract information from raw data.

The aim of this study is threefold. One is to describe some of the recent advances in this area. We focus on RLPM, on one hand, and on the brain as a prototypical example of a biological porous medium, on the other hand. We argue that ML algorithms, which have been developed to enhance the resolution of images of RLPM and other complex materials, can also be used to enhance the resolution of MIs, thereby enabling more accurate diagnoses. Thus, both communities can benefit from the achievements of each other in using ML approaches to analyze digitized images of their respective porous media. The second aim is to highlight that the same techniques developed for modeling of RLPM can be applied to MIs, particularly those of the brain. Finally, in some respects, the field of RLPM modeling is more advanced than the medical field dealing with biological porous materials, in that more rigorous analyses and highly realistic models have been developed, whereas modeling of biological systems is often characterized by empiricism and *ad hoc* regression. For example, pore-network modeling of various flow phenomena in RLPM is highly advanced, and many rigorous methods and models have been developed. Thus, it can be highly beneficial for the medical community working on biological porous media, as described here, to take advantage of this progress in their efforts to gain a better understanding of the systems of interest.

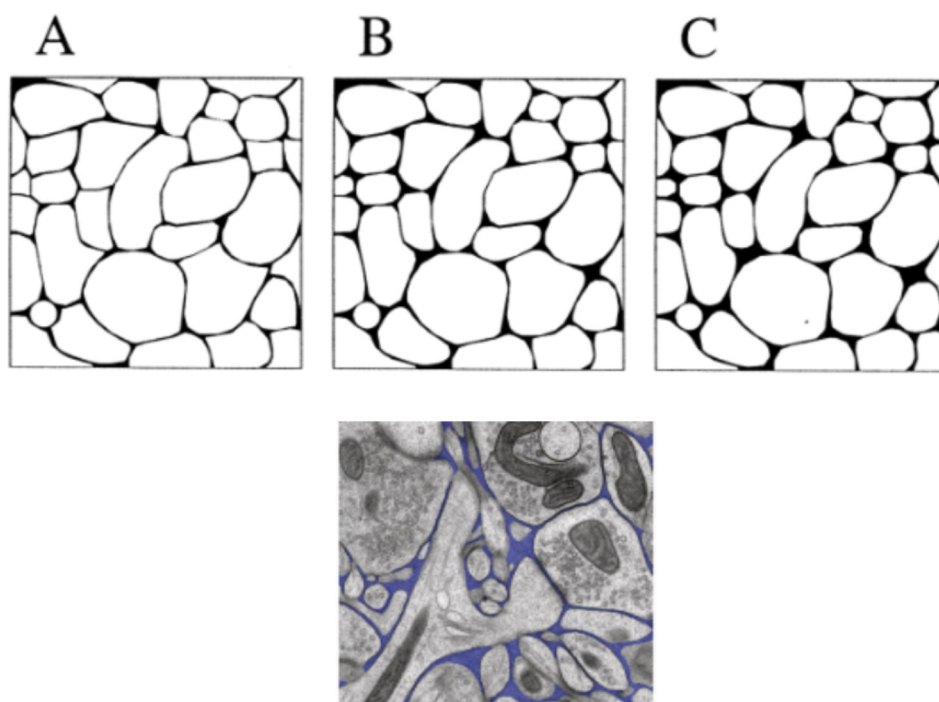


FIGURE 3

Brain extracellular space as an unconsolidated porous medium that mimics a two-dimensional section of neuropil structure shown at the bottom. The porosity is (A) 0.13; (B) 0.16, and (C) 0.18.

The connection between imaging of geomaterials and CT scanners used in the medical field is not entirely new and dates back nearly 40 years, when researchers at Shell Oil Company used medical CT scanners to characterize rock [22, 23]. What we propose in this study is, however, more comprehensive, in that we believe the medical community can take advantage of advances in the modeling and characterization of RLPM, while researchers in the area of RLPM can benefit from advances in instrumentation for medical imaging and from the way such images are interpreted and understood.

2 Classification of images

Classification of digitized images—in particular MIs and those for RLPM—is an important problem that, despite significant progress, is still being studied. The approach in both cases is essentially the same, although the goals may be different. In what follows, we describe some of the recent progress that has been made. It should be noted that the accuracy of all the models described below was tested not only by comparing their predictions with the existing data but also through various statistical measures, such as the coefficient of determination R^2 .

2.1 Rock-like porous media

Classification of images of RLPM is multifaceted. In one problem, one would like to distinguish pores of different sizes and

classify them into micro-, meso-, and macropores. Note, however, that there are multiple definitions of the three length scales, or respective ranges of pore sizes, so that those working on imaging of biological systems and geologists may not be entirely consistent in their interpretation of what the pore sizes represent. Many porous media contain microfractures, and therefore, another problem is *detection*, i.e., identifying microfractures and distinguishing them from large pores. One issue in both cases is a lack of extensive data for training the ML algorithm. Addressing this problem is relatively straightforward and is usually accomplished through *transfer learning* (TL), i.e., using NNs that have already been trained on other types of data related to porous materials that are relatively abundant and then fine-tuning them for the properties of interest. For example, [24] used CT images together with five well-known pre-trained ML models, namely, VGG-16, ResNet-50, InceptionV3 [25] (introduced by Google), DenseNet121 [26], and MobileNet [27], all of which represent deep convolutional NNs (CNNs), to generate accurate synthetic data for pore sizes and then used the NNs to classify images of soils based on their pore-size distribution, classifying the pores into various types mentioned above.

2.1.1 Detection

Several studies address the detection problem. One notable example is the work of [28], who developed an ML-based algorithm using eighteen images for thin sections of carbonate rock that were scanned at a resolution of 6.35 microns/pixel. To correctly classify the structure, only open-mode microfractures and pores were considered, and healed microfractures, usually referred to as

microveins, were ignored. The edges of all images were cropped to remove the blank slide edges, after which they were pre-processed by denoising and sharpening. Various methods can be used for denoising, such as those based on wavelet and curvelet transformations [29, 30] and other methods [31] that preserve the edges of the images. If, after denoising, parts of the images were not sharp, an unsharp mask filter was used to restore the sharpness. Thresholding was used to segment the pores that were filled with the blue epoxy. A complication was that microfractures that seemed continuous were actually fragmented into smaller segments after thresholding. To address the issue and obtain the correct microfracture connectivity, a device-independent 3D color space was used that accurately maps all perceivable colors, the result of which was then combined with the original segmented image after post-processing both images. Microporous matrix zones and microporous grains were segmented as macropores, and due to the sheer number of microporous zones, they were considered pores. Many smaller pores were poorly resolved, with their true shapes lost. To address the issue, [28] visually estimated the smallest pore size that could adequately be resolved, which in their images was 30 pixels in area. Thus, all objects below the threshold were removed from the images, while care was taken to ensure that microfractures were not removed since their number throughout the dataset was limited.

The binary systems were processed in Python for labeling and feature extraction. Feature selection was supervised and based on *a priori* knowledge of the features and their correlations. The features included area, perimeter, filled area (defined as the number of pixels in the object with holes filled), major and minor axis lengths (defined as the lengths of the two axes of the best-fitting ellipses), and a few others. To improve the accuracy of the ML approach, a manual approach was used to remove the outliers. The idea was to examine the data points, and if they were more than 10 standard deviations from the mean of both size and shape features, they were visually corroborated with their corresponding images before being considered outliers. The labeled dataset consisted of 400 pores and 400 microfractures, with the pores selected randomly, while the microfractures were selected manually. Secondary labels were added to each sampled object that pertained to the type of pore or microfracture. Four types of microfracture were identified, namely, straight, curvilinear, curved, and branching. Seventy percent of the labeled dataset was randomly selected for training, with the remaining thirty percent used for testing, while maintaining the same proportions of pores and microfractures in both sets.

Several supervised ML models were then tested. They included random forests (which combine several decision-trees with randomly distributed features to generate a majority vote for classification, which means that they mix several weaker learners into a single robust learner), variants of support vector machines (which work based on identifying the most effective boundary to separate different classes of data points), and the K -nearest neighbor algorithm. The hyperparameters of the various algorithms were then optimized. As the next step, learning curves, which graphically represent the degree to which the ML model learns the classification task on incrementally larger portions of a training dataset, were generated for the ML models in order to evaluate their stability and detect any overfitting. The training and resampling increments were set at ten percent of the training dataset. The accuracy of the

ML algorithms was assessed using the standard method, i.e., by identifying the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) and defining accuracy A as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN}.$$

In addition, sensitivity S of the algorithm is defined as follows:

$$S = \frac{TP}{TP + FN},$$

while specificity \mathcal{S} was taken to be,

$$\mathcal{S} = \frac{TN}{FP + TN}.$$

All the supervised ML models produced highly accurate predictions, within a narrow range of 93.64–94.63, implying that ML algorithms can accurately classify images of RLPM. Figure 4 presents an example of carbonate rock. [28] used two binary masks [a mask is a specific piece of data used for bitwise operations (such as AND and OR) on the bits, i.e., 1 and 0], namely, hue–saturation–brightness (HSB) masks and LAB masks. The HSB color space describes colors using three properties, namely, hue (representing a pure color), saturation (that varies from gray to vivid), and brightness (from black to bright color), and is a cylindrical model used in digital art. LAB (or CIELAB) color space is a model that describes colors with three values, namely, L for lightness, A for green-to-red, and B for blue-to-yellow.

Other variations of this basic model are described extensively by [32].

2.1.2 Segmentation

Another problem in the classification of images of porous media is *segmentation*, i.e., dividing the pores and minerals into separate phases, an important problem that also arises in multiphase fluid flow when one must not only distinguish the pores and minerals but also various fluid phases, along with the wettability of the pore space. Segmentation models represent classification models in which the task is executed for each pixel in an image. Segmentation of, for example, micro-CT images by ML algorithm began with very basic methods, such as random forest classification, which proved more successful than most deterministic segmentation approaches (see, for example, [33]). Random forest classification has been successfully implemented in several open-source and commercial software programs, such as Fiji (trainable WEKA segmentation [34]), Ilastik, Reactiv'IP SDK, and Dragonfly. Random forest classification is also the underlying method in many other software packages, where, for example, a higher-resolution SEM image is used to up-resolve the EDX-map image in electron microscopy [35, 36].

Deep NNs have also been used for classification, particularly segmentation, of RLPM. [37] developed an ML-based algorithm for the segmentation of images of RLPM. They used SegNet [38], a fully convolutional encoder–decoder network in which the encoder uses a CNN, but its fully-connected layers were removed in order to produce a low-resolution feature map of the input image. The decoder utilizes a similar architecture to generate a high-resolution feature map, which is then fed to a multiphase softmax layer of the CNN to classify it into a pixel-wise multiphase segmented output. Two SegNet networks were used. One was the standard network [38] with four encoders and four decoders, with each

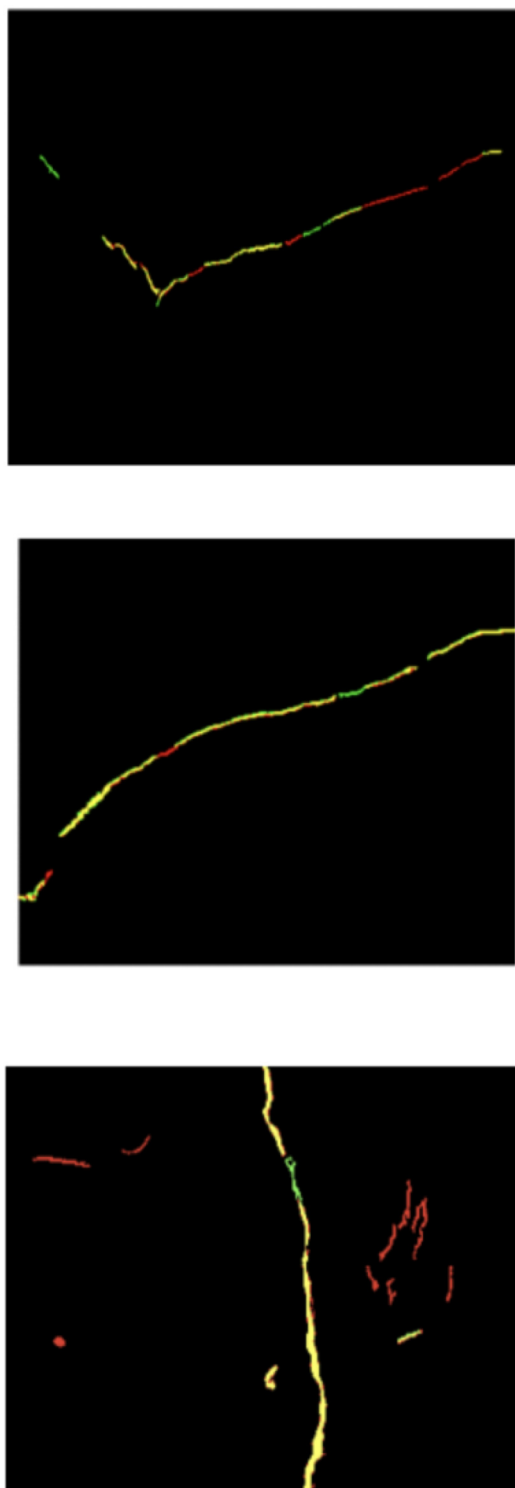


FIGURE 4
Composites of the HSB and LAB binary masks, segmenting the microfractures. Red represents the HSB binary mask; green is the LAB binary mask, while yellow indicates the union of the two. Overall, the HSB mask produces a stronger segmentation, but the LAB mask provides a notable boost in connectivity (based on [28]).

encoder/decoder having a $7 \times 7 \times 64$ convolutional/deconvolutional layer and a 2×2 max-pooling/un-pooling window and stride 2. The rectified linear unit (ReLU) activation function was used only in the decoder part. The second SegNet was similar to the standard version but featured a larger architecture with five encoder/decoder pairs, two convolutional/deconvolutional layers, and three convolutional/deconvolutional layers in the six central parts. Since, as mentioned above, massive amounts of data are needed for training the NNs, [37] used the cross-correlation simulation (CCSIM) method of [39–42] to generate a large number of synthetic images based on the original ones. The results were then used to train the autoencoder CNN and to carry out the segmentation of images of several Berea sandstone samples.

Segmentation was carried out by thresholding the values of the pixels, resulting in automatic segmentation. An example is shown in Figure 5b. It turned out, however, that two critical issues must be addressed for correct segmentation. As Figure 5a indicates, grain boundaries are brighter than grain surfaces, and thus, they can be misclassified. In addition, various minerals can be similarly classified, due to their close color values or intensities. To address these issues, [37] manually labeled each misclassified pixel as an “expert supervisor,” thereby resulting in a semi-automatic segmentation algorithm. Figure 5c shows the result of the semi-manual segmentation. Five phases were identified: pore space, quartz, K-feldspar, zircon, and other minerals that were mainly clays.

2.2 Biological porous media

Precisely, the same principles used for classifying images of RLPM can also be applied to MIs, for which two types of classification are of interest, as follows.

2.2.1 Exam and lesion classifications

In *exam classification*, the input to the ML algorithm is one or several images (exams), but there is a single output, such as “is the disease present or not?” Clearly, dataset is small, and therefore, a TL method must be used to meet the large-data requirement of deep NN training. Two TL strategies have been used. One, as discussed above, is based on using a pre-trained NN as a feature extractor, which has the advantage of not requiring the user to train a deep NN, thereby allowing the extracted features to be easily inserted into the existing image analyzers. The second approach fine-tunes the pre-trained NN on medical data. Fine-tuning trains the parameters of an already pre-trained NN on new data, which can be carried out on the entire NN or on only a subset of its layers. In CNNs, it is common to keep the earlier layers frozen as they are closest to the input layer and capture lower-level features, while subsequent layers often discern high-level features that are more directly related to the task on which the model is trained. In many studies, however, the TL strategy has been arbitrarily configured. [43] provided a comprehensive review of efforts to offer guidance for selecting a model and TL approaches for the classification of MIs. [44] and [45] fine-tuned a pre-trained InceptionV3 architecture on medical

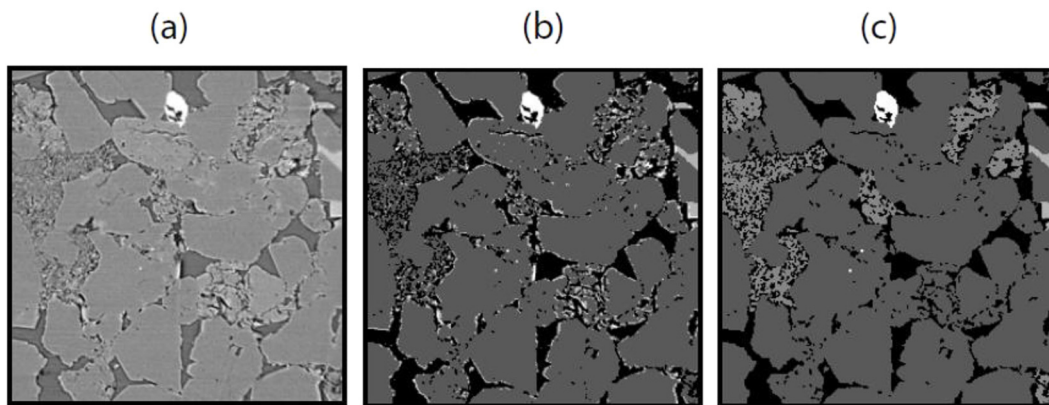


FIGURE 5

(a) Original CT image of a sandstone; (b) automatic segmentation and (c) semi-manual segmentation of the image in (a) [37]. In the right image, gray, light gray, white, semi-gray, and dark gray represent, respectively, quartz, ankerite, zircon, K-spar, and clay.

data and reported near-human expert performance. [46] reviewed properties of TL for medical imaging. On the other hand, [47] carried out computations to compare training from scratch with fine-tuning of pre-trained NNs. Their work showed that fine-tuning can be more accurate if a small dataset of approximately 10^3 images of skin lesions is available, although the computer experiments were conducted on too small a scale to allow any general conclusions to be drawn from the results.

The second type of classification of MIs is *lesion classification*, the medical version of the problem of object classification, in which one focuses on classifying a small part of an MI into two or more classes, such as nodule classification in chest CT. Accurate classification requires, however, both local information on lesion appearance and global contextual information on lesion location, which is typically not possible in generic deep learning (DL) architectures. To address this problem, multi-stream architectures, such as multiple CNNs, are used in a multiscale fashion, allowing all contextual information to be processed, regardless of the image size. Accurate lesion classification often requires using 3D information. Note, however, that images in computer vision are usually in 2D, and therefore, NNs pre-trained on such images cannot be used directly when 3D information is available. There are various approaches to addressing this issue. For example, [48] utilized a multi-stream CNN to classify points of interest in chest CT as a nodule or non-nodule, with up to nine patches with various orientations extracted from the candidate image used in separate streams and merged in the fully connected layers to obtain the final classification output. Note that the typical loss function used in classification is based on cross-entropy loss, which is given by

$$\mathcal{L}_c = - \sum_{i=1}^C y_i \ln p_i, \quad (1)$$

where C is the number of classes classified and y_i is the label, which is 1 when and only when $y_i = i$; otherwise, $y_i = 0$. p_i is the predicted probability that the model assigns the current image to class i . As for the location loss, the mean-squared error in position coordinates is typically used.

2.2.2 Detection

Similar to RLPM, the problem of detection is also important for MIs, where detection represents a more refined approach to image classification. This is because every pixel in an image is classified, and thus, the class balance can be skewed strongly toward the non-object class in a training dataset. While the majority of non-object samples are typically straightforward to discriminate, this prevents the ML method from focusing on the difficult datasets. As described above, the same problem arises in RLPM when, for example, one wishes to differentiate between microfractures and macropores. To address this issue, [49] proposed a method for selective sampling of data, in which datasets that are incorrectly classified are fed back to the NN more frequently to focus on difficult areas in the images, which, in their case, were retinal images. In addition, because classifying each pixel in a sliding window leads to orders of magnitude of redundant or useless calculation, CNN [50] can be an important aspect of an object detection approach. In the standard approach of using CNNs to classify each pixel in an image individually, patches of extracted data (the sliding window) around the particular pixel are fed into the network. The problem is that input patches from neighboring pixels have very significant overlap, and therefore, the same convolutions are computed too many times—a waste of computation time. The same problem arises in the detection of images of RLPM. The convolution and inner (dot) product are both linear operators; this implies that one can be expressed in terms of the other. Thus, if fully connected layers are rewritten as convolutions, then a CNN can take input images larger than those on which it was trained and produce a *likelihood map*, rather than an output for a single pixel. The result is a fully CNN.

Detecting patterns or landmarks in the human body often requires a spatial 3D dataset, which is not straightforward to use with NNs, representing a complex process. One way of addressing this issue is to represent the 3D image as a set of 2D orthogonal slices. For example, to identify landmarks on the distal femur surface, [51] used three independent sets of 2D MRI slices with CNNs and defined the spatial position of the landmark as the intersection of the three 2D slices with the highest classification output. A more advanced technique was developed by [52], who detected

regions of interest around anatomical regions, such as the heart and aortic arch, by identifying a rectangular 3D bounding box after 2D analyzing of the 3D CT volume. Note that, unlike many problems for which ML algorithms have been used, when it comes to the MIs that are about a specific problem, one may not have access to a considerable volume of data, and therefore, one often uses pre-trained CNNs to learn better feature representations [53]. Each landmark can also be represented, for example, by a Gaussian distribution, and a map of such landmarks [54] can be used as input to the CNN for training and subsequent prediction of the map. Another approach [55] is based on applying reinforcement learning to detect landmarks. To solve the difficult problem of direct detection and localization of landmarks and regions in 3D images, [56] partitioned 3D convolution into three 1D ones in order to detect carotid artery bifurcation in CT images.

Temporal data contained in, for example, a video, have also been analyzed for detecting landmarks. For example, [57] trained CNNs on the data contained in a video in order to detect up to twelve standardized scan planes in mid-pregnancy fetal ultrasounds. To locate the brain and spine in the scan plan, [58] utilized saliency maps. [59] developed a 4D (three spatial dimensions plus time) reconstruction method based on a deep CNN, dubbed 4D CINENet, for cases in which 3D Cartesian CINE imaging is undersampled, i.e., when insufficient data are available. Their NN was based on $(3 + 1)$ D complex-valued spatio-temporal convolutions and multi-coil data processing. They trained and evaluated the proposed CINENet on 3D CINE data from 20 healthy subjects and 15 patients with suspected cardiovascular disease and showed that CINENet performs much more accurately than iterative reconstructions in terms of visual image quality and contrast. In addition to CNNs, recurrent NNs (RNNs) and long short-term memory (LSTM)-RNNs have also been used with information contained in medical videos. An example is the work of [59], who utilized LSTM models to incorporate dynamic information contained in consecutive sequences in ultrasound videos to detect fetal standard planes, while [60] combined an LSTM-RNN with a CNN to detect the end-diastole and end-systole frames in cardiovascular MRI, also called cine-MRI, or the “gold-standard” for the assessment of cardiac function of the heart.

In general, detection models in MIs, including those described above, are typically one of the two types, namely, single-stage or two-stage models. A single-stage detection (SSD) model must operate in a timely manner, with a low runtime, in order to be efficient. The first version of the algorithm, dubbed [61] YOLO (“You only Look Once”), formulated object detection as a regression problem. To do so, it discretized the image into a grid or mesh, instead of initializing the detection frame, thereby generating spatially separated grid blocks or bounding boxes and the associated class probabilities. A single NN was then used to predict bounding boxes and class probabilities directly from the full images in a single evaluation. Since the entire detection algorithm was based on a single NN, it could be optimized end-to-end directly on detection performance. However, if multiple objects fall on the grid at the same real time, YOLO version 1 becomes ineffective. The second version of YOLO, YOLO9000, developed by [62], addressed this shortcoming by adding to the grid the number of blocks to be selected and fine-grained features through the passthrough layer. Using a convolution operation, instead of a pooling operation, YOLOv3, also developed

by [63], made an “incremental improvement” (in the authors’ words) by incorporating a residual structure and a feature fusion strategy, which not only improved its generalization performance but also reduced the number of parameters. YOLOv4, developed by [64], utilized data processing, a backbone network, an activation function, a loss function, and other strategies in order to speed up the required computation, turning it into an efficient detection model. Another detection approach, the single-shot multibox detector (SSD), developed by [65], also discretized the output space of bounding boxes into a set of default blocks with various aspect ratios and scales per feature map location. To make a prediction, the deep CNN generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Moreover, the proposed approach combines predictions from multiple feature maps with varying resolutions in order to handle objects of various sizes effectively.

A prototype of a two- or multi-stage detection algorithm is that of [66]. It consisted of two key ideas: (a) applying high-capacity CNNs to bottom-up region proposal networks (RPNs) in order to localize and segment objects and (b) carrying out supervised pre-training for an auxiliary task when labeled training data are scarce, followed by domain-specific (or region-specific) fine-tuning (see above), which improves the performance significantly. An RPN [67] takes an image of any size as input and produces a set of rectangular object proposals, i.e., where an object could be, each with an objectness score, which is a confidence measure that is used in object detection models. It indicates the probability that a proposed region of interest contains an object, irrespective of its specific type. Because [66] combined RPNs with CNNs, they dubbed the method R-CNN, in which regions are represented using CNN features. An improved and computationally faster version of R-CNN, dubbed Fast R-CNN, was proposed by [68]. Since many detection algorithms that use deep CNNs require a fixed-size input image, [69] developed a method that no longer needed the constraint. In their model, dubbed Special Pyramid Pooling-Net (SSP-net), the last pooling layer (after the last convolutional layer) was replaced with a spatial pyramid pooling layer with a large number of bins. In each spatial bin, the approach pools (max pooling was used) the responses of each filter. The outputs of SPP are kM -dimensional vectors, with M and k , being respectively, the number of bins and the number of filters in the last convolutional layer. The vectors have fixed dimensions and are the input to the fully connected layer of the CNN. Thus, the input image can be of any size, which not only allows arbitrary aspect ratios but also arbitrary scales. Note that the typical loss function in both methods includes terms corresponding to classification and location loss functions. The former is a multivariate cross-entropy loss, given by Equation 1. As for the location loss, the mean-squared error in position coordinates is usually used.

2.2.3 Segmentation

Similar to RLPM, segmentation of MIs is also an important problem. In the context of MIs, segmentation of the images is defined as the identification of a set of voxels (pixels) that constitute either the contour or the interior of the organs of interest in the images. Deep learning-based models of segmentation typically use the architecture of the encoder-decoder of SegNet (see above). As discussed above, features at a high level of abstraction are obtained in the encoder by convolution and downsampling, while the feature

map is recovered to the original size of the image by convolution and upsampling. The output is the segmentation of the image. The U-Net architecture, which is widely used for medical images, combines an equal number of upsampling and downsampling layers, with so-called skip connections between corresponding convolution and deconvolution layers. Skip connection, also referred to as residual connections, bypasses one or more layers by adding the input of a block directly to its output, thereby forming a shortcut, or skipping path, that helps mitigate the vanishing gradient problem in minimizing the loss function. Skip connection enables training of much deeper NNs since it makes it possible for gradients to flow more easily through an NN and preserve important features from earlier layers. This concatenates features from the contracting and expanding paths so that the entire image is processed in a single forward pass, directly generating a segmentation map directly while taking into account the complete content of the image. It was shown by [70] that a complete 3D segmentation can be achieved if the U-Net is provided with a few 2D annotated slices from the same 3D image. [71] improved this basic scheme. U-Net and its variants are now the standard NNs for segmentation of lesions. Let us describe a particular variation of U-Net that has attracted considerable attention.

[72] developed a deep learning-based segmentation method for MIs, dubbed nnU-Net, which configures itself automatically and carries out multiple tasks, including preprocessing, network architecture, training, and post-processing for any new task in the biomedical domain. The configuration of nnU-Nets consists of two blocks per resolution step in both the encoder and decoder, with each block consisting of a convolution, followed by instance normalization and a leaky ReLU, and is shown in Figure 6. The architecture of nnU-Net is the same as that of the standard U-Net, but with minor adjustments. In order to enable large patch sizes, nnU-Net's batch size is as small as two. Batch normalization is typically used in CNNs for speeding up or stabilizing the NN training. It does not, however, perform well when the batch sizes are small. Therefore, [72] utilized instance normalization [73]. In a CNN, batch normalization normalizes all images across the batch and spatial locations. In instance normalization, on the other hand, each element of the batch is normalized independently. [72] used the leaky ReLU (with a negative slope of 0.01), instead of ReLU. The training was performed with deep supervision, and additional auxiliary losses were added to the decoder at all but the two lowest resolutions, allowing gradients to propagate deeper into the network and facilitating the training of all the layers. [72] performed downsampling using strided convolution, while upsampling was carried out using transposed convolution. To balance the requirements for performance and memory, the initial number of feature maps was set to 32, and doubled (halved) with each downsampling (upsampling), while to limit the size of the final model, the number of feature maps was limited to 320 and 512 for 3D and 2D U-Nets, respectively.

[72] trained all networks for 10^3 epochs, with one epoch defined as an iteration over 250 mini-batches. Stochastic gradient descent with an initial learning rate of $l_r = 10^{-2}$ was used for learning the weights of the NNs. The loss function was the sum of cross-entropy (see above) and Dice loss. Also known as the Dice-Sørensen coefficient, the Dice loss is a statistical measure used to quantify the similarity between two samples. For each deep supervision output, a

corresponding downsampled ground-truth segmentation mask was used in order to compute the loss. The objective function of the training was the sum of the losses at all resolutions. Samples for the mini-batches were selected from random training cases. Class imbalance occurs when some types of data are much more common than others that are exceedingly rare. For example, most voxels in an MI belong to the non-diseased class. As a result of class imbalance, NNs and other ML algorithms are often trained in class-balanced settings. To address the issue, [72] implemented oversampling. Two-thirds of the samples were taken from random locations within the selected training case, while the remaining one-third of the patches were ensured to contain one of the foreground classes present in the selected training sample.

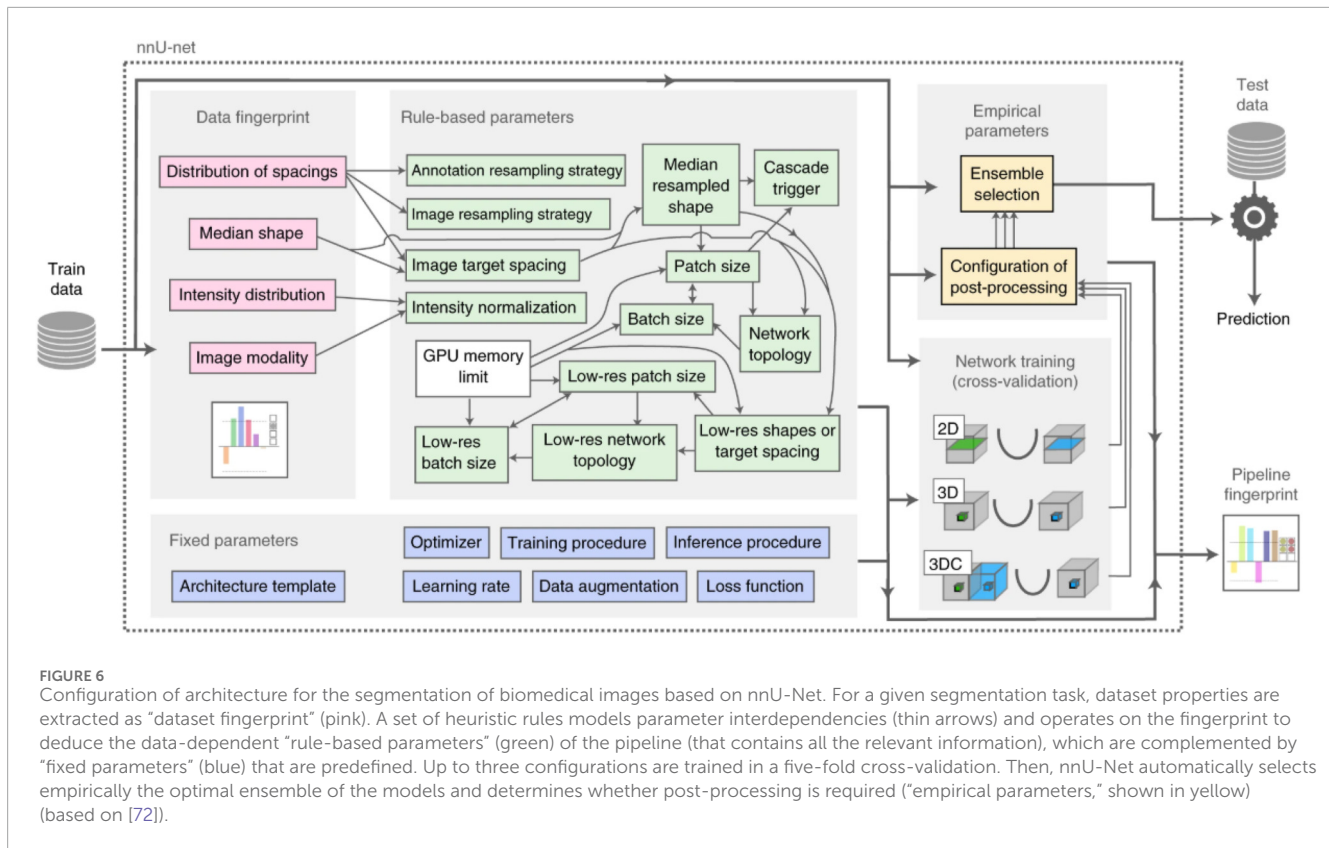
Accurate segmentation also requires global and local context, which is why multi-stream networks with distinct scales or non-uniformly sampled patches are used. For example, [74] developed a deep 3D CNN with eleven layers for the segmentation of brain lesions. Since intensive computations are needed for processing 3D medical scans, [74] developed an efficient training scheme that joined the processing of adjacent image patches into one pass through the NN, while also adapting to the inherent class imbalance present in the data, i.e., most voxels in an MI belonging to the non-diseased class. The class imbalance problem can be addressed by defining a loss function as a weighted combination of sensitivity and specificity (see above), whereby a larger weight is attributed to specificity to make it less sensitive to the data imbalance. To incorporate both local and global information, a dual pathway architecture was used by [74], which simultaneously processed the input images at multiple scales while using a 3D fully connected conditional random field to effectively remove false positives and to carry out post-processing of the network's soft segmentation. [75] trained CNNs using non-uniformly sampled patches to achieve a wider coverage of the sampled regions, which captured more contextual information, proving very useful for the analysis of medical images.

3 Enhancement of the images of porous materials

Enhancement techniques are used to refine images of porous materials so that their important features become easier to understand and can be detected by automated image analysis systems. At a fundamental level, image enhancement represents the mapping of one image onto another, which is not necessarily one-to-one, meaning that two seemingly different images can be transformed into the same or similar output images after enhancement. The process is not without problems because the enhancement of some features in the images may produce undesirable effects. In addition, important information may be lost.

3.1 Rock-like porous media

The use of ML algorithms in problems involving porous media can be limited by the lack of large datasets required for training the algorithms. One way to address this problem is through the enhancement of images of porous media, which allows for the



expansion of the dataset and more accurate estimation of the physical properties of the media. However, the application of ML algorithms to problems involving porous materials and media, particularly their images, initially encountered challenges. For example, pixel values in an image must first be transformed into a feature vector that can be detected by the ML method. After deep-learning methods were developed in early 2006, trained using multiple levels of representation, progress began to accelerate. Raw data can then be used as input to accomplish important tasks such as the detection or classification of certain features in datasets and image enhancement.

3.1.1 Super-resolution

Super-resolution (SR) [76, 77] is a classical ill-posed problem studied by computer scientists, which aims to reconstruct high-resolution (HR) images from low-resolution (LR) images and directly enhance image resolution beyond hardware limitations. The problem is ill-posed because the solution that the method produces is not unique. Thus, coarse images taken across large fields of view (FOVs) are artificially enhanced to the resolution needed for carrying out accurate modeling, circumventing traditional trade-offs in analyzing multiscale porous media. In the classical reconstruction problem, one uses a given dataset for a porous medium—images of the pore space in this case—in order to develop a model of the pore space that not only honors the data but also, when used in numerical simulation of various phenomena in the pore space, provides accurate estimates for the properties of the porous medium [78]. Super-resolution generates an image from a single LR image using a feedforward CNN, referred to as the

generator \mathcal{G} that consists of multiple layers with the usual weights w_L and biases b_L , which, as usual, are determined by optimizing a loss function. If the LR image is 2D, it can, as the input, be described as a tensor of shapes (n_x, n_y, n_c) , where (n_x, n_y) and n_c are, respectively, the dimensions of the image and the number of color channels present. In 3D, a third dimension is added so that an image of size (n_x, n_y, n_z, n_c) is super-resolved. A scale factor s is typically used so that one obtains, for example, an SR image of size (sn_x, sn_y, sn_z, n_c) . Generative adversarial networks (GANs) and CNNs have been utilized with SR for image enhancement.

[79] proposed an image SR using a deeply-recursive CNN with a very deep recursive layer with up to 16 recursions. Increasing recursion depth improves the performance of the algorithm without introducing new parameters for additional convolutions. However, the difficulty with their method is that training the NN using the standard gradient descent was very complex, due to exploding or vanishing gradients during the optimization of hyperparameters. [80] proposed a very deep fully convolutional encoding–decoding architecture for restoring images, such as denoising and SR. Their NN consisted of multiple layers of convolution and deconvolution operators that learned end-to-end mappings from corrupted images to the original images. The convolutional layers acted as the feature extractor, capturing the abstraction of image contents while eliminating noise and corruption. Deconvolutional layers were used to recover the details of the image. Symmetrically linked convolutional and deconvolutional layers with skip connections were used. The reason for using skip connection was that, as discussed above, the performance of deep NNS deteriorates with increasing the depth

of its architecture, which is referred to as *the degradation problem*. Skip connections were introduced to address this problem by, for example, gates that controlled and learned the flow of information to deeper layers. Using the skip-layer connection, the training converged much faster because it allowed the data to be back-propagated to the bottom layers directly and, thus, addressed the problem of gradient vanishing, which facilitated training deep NNs. In addition, the skip connections transferred image details from the convolutional to the deconvolutional layers, which is useful for recovering the original image.

[81, 82] used CNNs to super-resolve an image with an NN that had three to five activated convolutional layers. They applied the algorithm to a bicubically upscaled LR image to recover the HR details by a direct mapping. The original architecture, which was dubbed super-resolution CNN (SRCNN), was improved iteratively with deeper, more complex NNs that improved the mapping from LR to HR images. Integrated LR-to-HR SRCNNs do not require bicubic upscaling as a processing step because the computational cost for a fully sized input image increases. [82] removed the deconvolutional layers at the end of the NN and replaced them with subpixel convolution in order to reduce checkerboard artifacting [83, 84]. The problem of checkerboard artifacts arises when NNs use a deconvolution operation that allows them to use every point in the small image to “paint” a square in the larger one. However, deconvolution can easily have “uneven overlap” that puts more of the “paint” in some places more than others, particularly when the kernel size—the output window size—is not divisible by the stride, the spacing between points. When that happens in 2D, the uneven overlaps on the two axes multiply together and generate a characteristic checkerboard-like pattern of varying magnitudes. Deeper networks produce improved results using the skip connection that adds outputs from shallow layers to deeper layers, which preserves important shallow feature sets and improves gradient scaling.

[85] used SRCNNs for image enhancement, using DeepRock-SR [86], a diverse compilation of raw and processed micro-CT images in 2D and 3D. The architecture that they used was based on the enhanced deep SR (EDSR) developed by [87] and the SR-Residual Net [88] NNs. The EDSR encompasses a sequence of convolutional layers, residual blocks, and an upsampling module, which also alleviates the problem of gradient vanishing. [85] retained the overall EDSR architecture but replaced the ReLU activation layers with parametric ReLU (PReLU) layers from SR-ResNet in order to avoid using batch normalization layers in SRCNNs, which have been shown to be detrimental to training, while the PReLU layers provide some improvement in the performance for a slightly increased computation [89]. The EDSR network that [85] used had convolutional layers of kernel size 3 with 64 filters over 16 residual layers and skip connections throughout the layers prior to upscaling. The network was set up in both 2D and 3D forms, with the only difference being the extra deep dimension that the convolutional kernels had for the 3D case; otherwise, the architectures of the 2D and 3D NNs were structurally identical. The loss function was defined as the pixel-wise mean of the sum of the differences between the generated SR and the original “ground-truth” HR image:

$$\mathcal{L} = \frac{\sum_i |SR_i - HR_i|}{\sum_i i}, \quad (2)$$

which presumably leads to more accurate results in a shorter time. The modified EDSR was trained for 100 epochs at 1,000 iterations per epoch with mini-batches of 16 cropped 192×192 images in 2D and batches of 4 cropped of size $96 \times 96 \times 96$ in 3D using the loss function defined in Equation 2 and a learning rate, $l_r = 10^{-4}$ using the Adam optimizer. The total number of trainable parameters for the 3D case was approximately 198 million, as opposed to approximately 102 million for the 2D case. At the end of each epoch during the training, which consisted of 1,000 mini-batch iterations, 1,200 full-size 500×500 images from the 2D dataset and 300 cubes of $100 \times 100 \times 100$ images from the 3D dataset were used to validate the model.

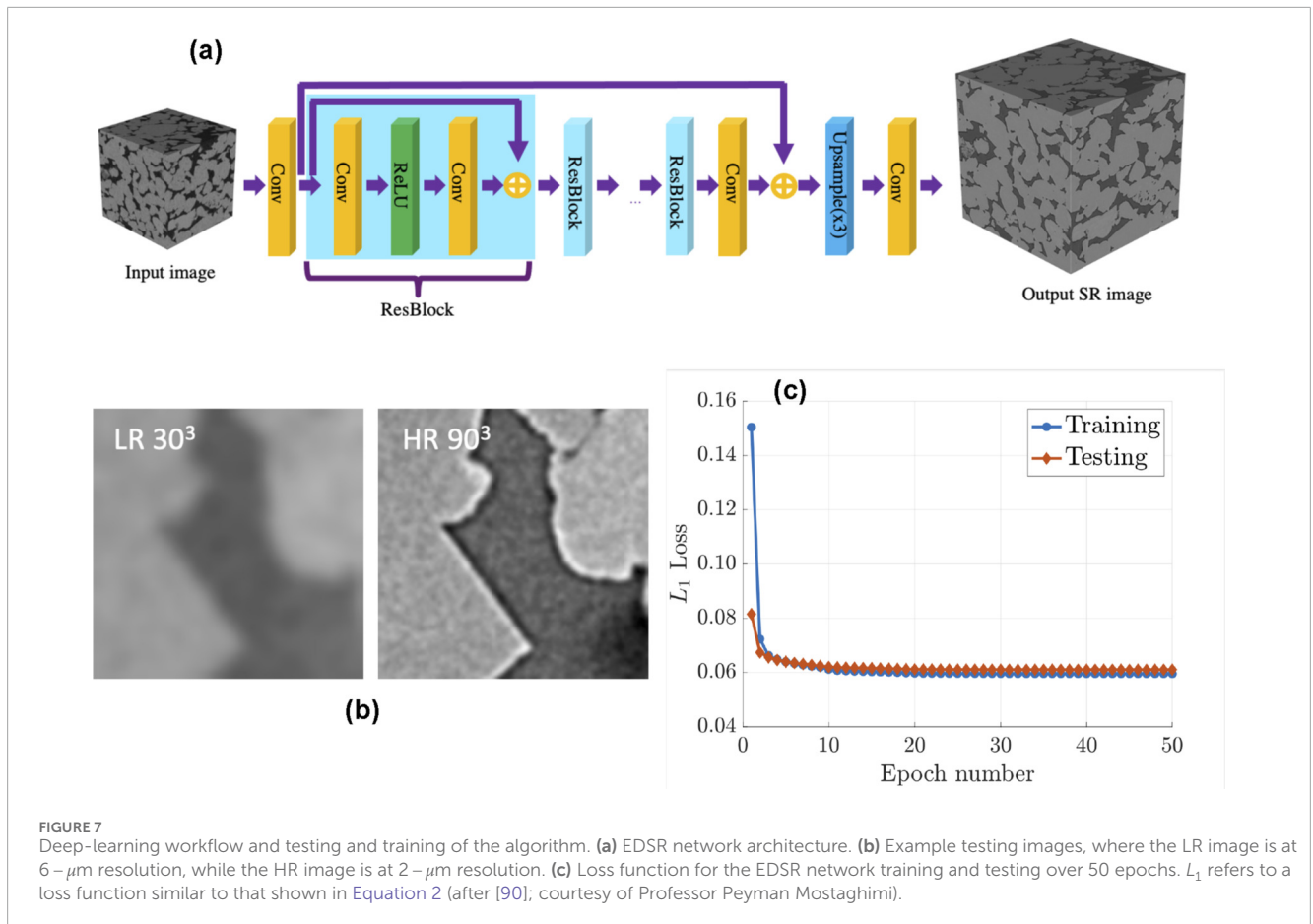
To further test the method, particularly validating it with data that are for quantities other than those related to the quality of the SR images, such as datasets for fluid flow and transport in the pore spaces represented by the images, [90] used SRCNNs for enhancing images of porous samples using experimental data, which were for two distinct Bentheimer sandstone cores of diameter 1.25 cm and length 6–7 cm, with varying centimeter-scale stratification, and consisted of $6\text{-}\mu\text{m}$ -resolution images of the entire cores in dry, brine-saturated, and partially saturated states during steady-state two-phase flow experiments. The dataset was enhanced by carrying out X-ray imaging of the cores in their dry state at varying resolution, using optical magnification, which generated paired data in four subvolumes at 2-, 6-, and $18\text{-}\mu\text{m}$ resolution. The paired data from core 1, subvolume 1, were used for training, while the other three subvolumes were utilized for validation. Image patches were extracted from the LR and HR data with sizes 30^3 and 90^3 voxels, respectively, generating 8,000 patches, from which 6,400 patches were used for training and 1,600 patches for validation. The patch size was selected to cover the largest typical pores present in the Bentheimer samples. A 3D EDSR model was again used. Moreover, the number of residual blocks and filters (12 residual blocks and 24 filters in each convolutional layer) was reduced to improve computational efficiency and mitigate overfitting to artifacts apparent in the HR micro-CT images. The Adam optimization algorithm was used for training with a batch size of 32 and a total of 50 epochs, selected after some preliminary computations. The learning rate was initially 10^{-5} and was decreased tenfold every 10 epochs to consolidate the training process. The loss function utilized was defined as follows:

$$\mathcal{L} = \sum_{i=1}^n |CT_{i,HR} - CT_{i,SR}|. \quad (3)$$

Equation 3 is similar to Equation 1. Here, $CT_{i,HR}$ and $CT_{i,SR}$ are the pixel grayscale CT values for the HR and generated SR images, respectively. Convergence was achieved after approximately 10 epochs. After training, the EDSR network was utilized to generate SR images with a threefold increase in resolution from any input LR images. The entire algorithm is shown in Figure 7. The EDSR was then used to generate the corresponding SR images for the unseen LR images from those samples that were not used in the training.

3.2 Biological porous media

Enhancement of images to higher resolutions, given images with lower resolution, in order to obtain more data with good



accuracy, is an important problem in medical physics that has been studied for a long time. The ML-based algorithms that were described above for enhancing images of RLPM can also be used in the context of MIs. [91] utilized multi-stream networks to reconstruct high-resolution cardiac MRI from one or more low-resolution 3D MRI images, precisely the same problem that was studied by [92] in the context of RLPM, which is described in the next section. Enhancement of medical images also has other applications. For example, one can not only recover missing spatial information but also utilize the enhanced image to infer advanced MRI diffusion parameters from limited data, as demonstrated by [93]. Comprehensive recent reviews are provided by [94] and [95].

4 Data augmentation and reconstruction of porous materials

As already mentioned, the use of ML algorithms for analyzing images and extracting information from them may be impeded by the fact that the number of available images may be severely limited, while training any ML algorithm essentially requires extensive data. In such cases, data augmentation is used, which, in the present context, refers to expanding the database by generating synthetic data based on the available images. One way to do this is through the classical problem of reconstruction,

which is defined formally as, *given a limited amount of data for a porous medium, how can one reconstruct its realizations that not only honor the data but also provide accurate predictions for its properties?* It should be clear that, unless one copies a 3D image of a porous medium, one can never reconstruct its exact replica. In other words, any reconstruction method can only generate plausible realizations of the original image whose accuracy depends on the reconstruction procedure and the amount of available data. This also implies that no reconstruction method can provide a *unique* solution to the problem. Several approaches to the reconstruction problem have been comprehensively reviewed and discussed by [78]. In this section, we describe two ML-based algorithms to address the problem of reconstruction of images of porous materials, both of RLPM and biological types. Both methods have been proven to be accurate.

4.1 Rock-like porous media

One ML-based approach to the reconstruction of porous media uses generative adversarial networks (GANs), developed by [96, 97]. Assume that a digital image \mathbf{DI} is a sample of a real probability distribution function (PDF) of the images f_d , of which we have a number of sub-domains that are extracted from \mathbf{DI} without any overlap. Every GAN consists of a pair of NNs, a *generator* G_θ , defined

by its parameter θ that generates realizations of the unknown PDF $f_d(\mathbf{DI})$, and a discriminator D_ω , with ω being a parameter that tries to distinguish between the realizations of the training set and the synthetic ones that are generated by G_θ . The latter learns to classify inputs as either real or false, whereas the former generates sufficiently realistic fake inputs to the discriminator so that they are classified as real. For example, one feeds random noise into a generator that is trained to transform the noise into data that closely mimic the real data from the training set. The discriminator is trained to distinguish the actual training dataset from the generated data. As the training of the two NNs progresses, the discriminator becomes more accurate at identifying the fake data, whereas the generator improves its ability to generate fake data for deceiving the discriminator. Thus, GANs are capable of generating realizations from an arbitrary PDF without any restriction on its form, which is the feature that makes them a powerful method for reconstruction. This is particularly true when the input for reconstruction is an image. The generator performs a mapping from a random prior \mathbf{Z} onto the domain of the image, defined as follows.

$$\mathbf{Z} \sim \mathcal{N}(0, 1)^{d \times 1 \times 1 \times 1}, \quad (4)$$

$$G_\theta: \mathbf{Z} \rightarrow \mathbb{R}^{1 \times n \times n \times n}, \quad (5)$$

In Equation 5 \mathcal{N} represents the normal distribution, while d in Equation 6 is the dimension of \mathbf{Z} , with n being the linear size of a subdomain extracted from the original image. $D_\omega(\mathbf{DI})$ assigns a probability to an image \mathbf{DI} from a sample drawn from f_d , the distribution of the true data: $D_\omega: \mathbb{R}^{1 \times n \times n \times n}$. Values close to 1 represent a high probability of being a sample of $\mathbf{DI} \sim f_d(\mathbf{DI})$. Both the generator and discriminator must be optimized, which can be carried out by a variety of methods. One can, for example, represent both $G_\theta(\mathbf{Z})$ and $D_\omega(\mathbf{DI})$ using differentiable NNs with parameters θ and ω . The optimization process is carried out in two steps:

- i. First, the discriminator is trained in order to maximize its ability to distinguish between real and fake samples. Training is carried out by a supervised deep-learning algorithm using the known real samples (label 1) and the realizations generated by G_θ (label θ). The loss (cost) function to compute the misclassification error is given by:

$$H(\mathbf{y}, \mathbf{y}') = - \sum_i [y_i \log(y'_i) + (1 - y_i) \log(1 - y'_i)], \quad (6)$$

Equation 6 defines the loss function as a cross-entropy in which \mathbf{y}' contains the output probability assigned by the discriminator to each element of a given mini-batch of samples. Thus, $H(1, \mathbf{y}')$ is optimized for each mini-batch of the real images, whereas $H(0, \mathbf{y}')$ is optimized for all the fake samples. The error is back-propagated while keeping the parameters of the generator constant.

- ii. Next, the generator is trained in order to maximize its ability to “deceive” the discriminator into misclassifying the images produced by the generator as real images. Training is carried out by computing the binary cross-entropy of the discriminator’s output on a mini-batch sampled from the generator $G_\theta(\mathbf{Z})$, requiring that the created labels be close to 1, thereby computing $H(1, \mathbf{y}')$, which is optimized by varying the

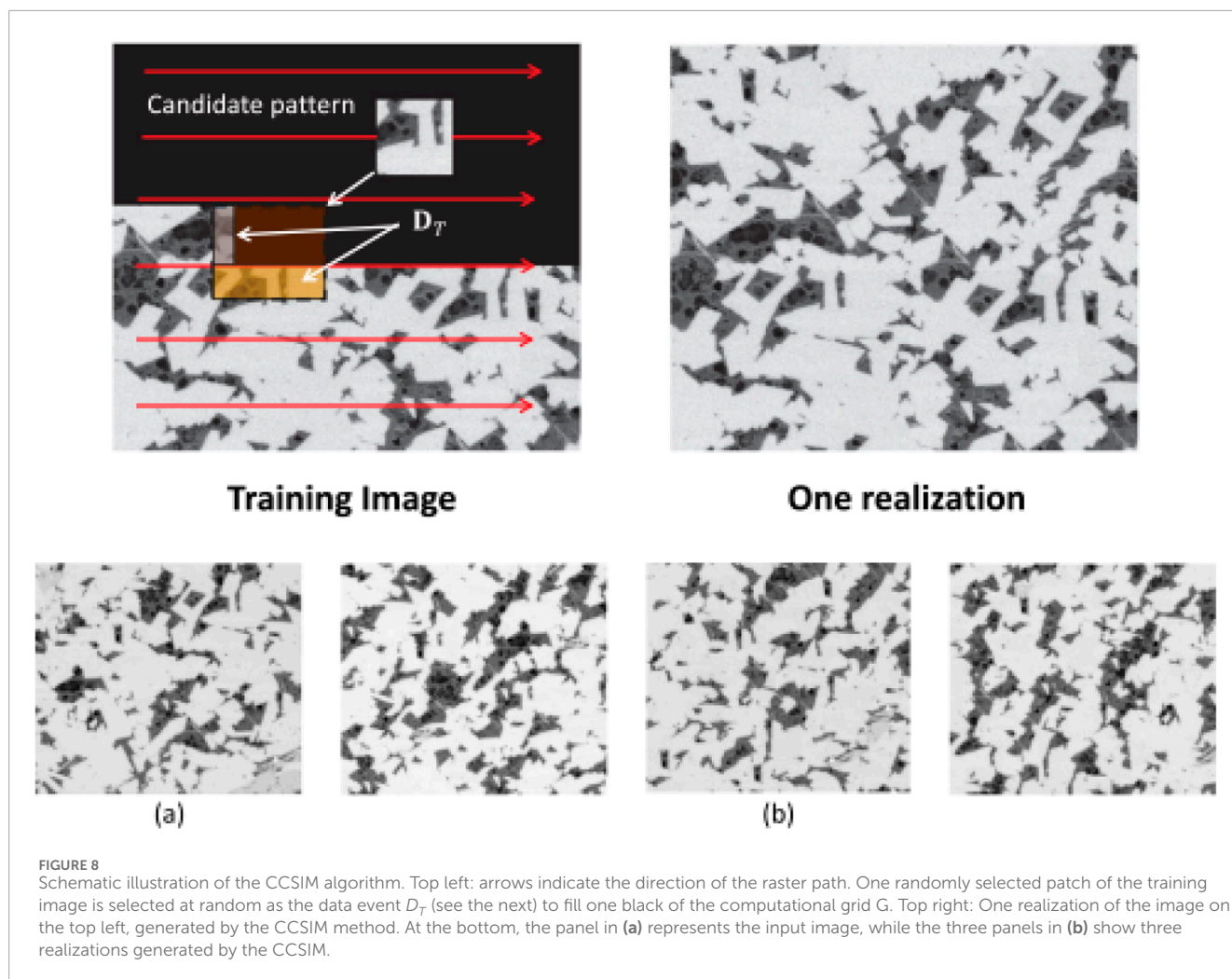
parameters of the generator while keeping the parameters of D_ω constant. The computations required for the optimization are intensive and cannot be performed by a single processor for large samples. For 3D images, the memory requirements are also substantial. [98, 99] used GANs to reconstruct a variety of RLPM.

By combining an SRCNN, representing the generator, with an image classification network as the discriminator, an SRGAN is formed. Although SRGANs produce features that appear realistic, due to pixel-wise mismatch, the accuracy of the generated SR images is lower than that of SRCNN. Since micro-CT images usually contain significant amounts of image noise and texture that manifest themselves as high-frequency features, they are also inadvertently recovered. According to [100], SRCNNs usually suffice because they contain some form of intrinsic noise suppression while maximizing edge recovery. [85] used the same dataset, DeepRock-SR, with SRGAN and compared its performance with that of SRCNN. Figure 7 shows the comparison of the results with those produced by SRCNN and BC-generated images.

The second approach is the CCSIM method [39–42], an ML approach without using any NN that represents the digital image \mathbf{DI} using a computational grid \mathbf{G} , partitioned into overlapping blocks of sizes $T_x \times T_y$, where \mathbf{G} and \mathbf{DI} have the same sizes. The neighboring blocks share overlapping regions \mathbf{OL} with sizes $\ell_x \times \ell_y$ (see Figure 8). One then begins from a corner block of \mathbf{G} (or any other block) and visits each grid block along a 1D raster path. For each grid block, a pattern of heterogeneity from \mathbf{DI} is selected at random and inserted in the visiting block. The inserted pattern is referred to as the *data event* \mathbf{D}_T , with the word “event” implying that the inserted pattern of heterogeneity in the block may change again. Then, the next pattern is selected based on the similarity between the neighborhood points and \mathbf{DI} , meaning that, instead of considering all the previously constructed blocks (by filling them with patterns of heterogeneity from \mathbf{DI}), only those in the neighborhood of the current blocks are used for the calculations. Next, the similarity between, or closeness to, the neighboring blocks and \mathbf{DI} is quantified based on the a cross-correlation function that represents a convolution between \mathbf{DI} and $\mathbf{D}_T(x, y)$:

$$\psi(i, j; x, y) = \sum_{x=0}^{\ell_x-1} \sum_{y=0}^{\ell_y-1} \mathbf{DI}(x+i, y+j) \mathbf{D}_T(x, y), \quad (7)$$

which represents a convolution of $\mathbf{DI}(x, y)$ with $\mathbf{D}_T(x, y)$, where $i \in [0, T_x + \ell_x - 1)$ and $j \in [0, T_y + \ell_y - 1)$. Thus, an overlap region of size $\ell_x \times \ell_y$ between two neighboring blocks and a data event \mathbf{D}_T are used to match the patterns in the \mathbf{DI} . The overlap region contains a set of pixels or voxels selected from the previously constructed blocks and uses them in Equation 7 to identify the next pattern of heterogeneity. For Euclidean distance (difference) between the constructed block and the data to be minimum, $\psi(i, j; x, y)$ must be maximum or, in practice, exceed a preset threshold. After calculating $\psi(i, j; x, y)$ and selecting those patterns for which $\psi(i, j; x, y)$ exceeds the preset threshold, one of the acceptable ones is selected at random and inserted in the block currently being visited in \mathbf{G} . The process is repeated until all the blocks of the grid \mathbf{G} have been reconstructed. As a rule of thumb, the neighboring regions might have an overlap of size of approximately



1/5–1/6 of the size of the blocks. Large grid blocks increase the computations as computing $\psi(i, j; x, y)$ requires considering many points, whereas small regions may cause discontinuity in the patterns. The method has been shown to successfully reconstruct models of various complex 2D and 3D porous media based on their images.

[92] proposed a method for enhancing images of porous materials and computing its effective properties (such as their permeability) when only a limited number of HR images—as few as one—are available. The method is based on data augmentation and generating a large and diverse dataset for the training of a CNN. [92] used the CCSIM algorithm to generate a large and diverse dataset to be used for training the CNN, and thus, they dubbed their approach a hybrid stochastic-CNN (HS-CNN) method. They developed their approach for shales, one of the most complex porous media, but the method is applicable to any type of porous material. They used a limited number of images—30 2D images with a size of 500×500 —with the CCSIM algorithm to generate 2,000 plausible realizations of the same, which were then used as the training images with the CNN. The size of 500 of the output realizations was 1000×1000 , with the remaining having the same size as the original images. The success of any learning algorithm

depends heavily on the number and diversity of the images that are used for its training. Thus, to diversify, the 2,000 training images were transformed using various filters, such as the Gaussian noise, scale, crop, rotation, and flip to increase the diversity between the training datasets. The CNN uses the high-quality training data and extracts their differences with the upscaled images of the same set, using a residual learning strategy. The training images all had high resolution, whereas the image at hand that required enhancement was a LR one, i.e., one with a smaller size. Therefore, the target image was enlarged, i.e., upscaled, to the size of the training images by using an interpolation method with bicubic functions. Then, the CNN learned iteratively how to estimate the residuals. After the training was complete, i.e., after the CNN learned how to estimate the residuals, the high-quality image was reconstructed by adding the original enlarged image to the estimated residuals. Therefore, the upscaled images were used as the input, whereas the estimated residuals represented the output. The CNN that [92] used had 22 individual layers that learned mapping of HR training images onto the LR input image. The utilized images were similar in their content but differed in their details. To reduce the computational time, 64 random patches of size 41×41 from the training data were selected from each training image and used in the NN, rather than utilizing

the full-size images. While such patches can jeopardize the capture of large-scale structures in the images, the use of the aforementioned filters—particularly, the scaling operators, ensured that the large-scale structures were still accounted for. The constructed small patches were then fed to the CNN over several iterations.

The LR image represented the first layer. The next layer was the convolution layer that contained 64 filters of size 3×3 , thereby assigning one filter to each patch. The performance of the CNNs improves with increasing the number of filters, but the training time also increases. Choosing a smaller filter size is usually preferred in the sense that it reduces the computational time, but it may also result in missing the large-scale structures of the image. On the other hand, larger filters complicate the training process. Thus, the optimal filter size should be decided *a priori* or selected by trial and error. Zero padding was also used in order to generate feature maps with the same input layer. All the convolution layers, except the last one, contained the ReLU activation function. The initial weights were assigned randomly and optimized during the learning. The first layer applied the Softplus, or SmoothReLU, function as the rectifier to the input values without changing their depth information, given by Equation 8:

$$\mathcal{A}(x) = \log[1 + \exp(x)]. \quad (8)$$

The last layer of the CNN had a single filter of size $3 \times 3 \times 50$, followed by a regression layer that calculated the loss function defined by Equation 9, given by:

$$\mathcal{L} = \frac{\sum_{i,j,k} [y_{\text{Resid}}(i,j,k) - y_{\text{HS-CNN}}(i,j,k)]^2}{\sum_{x,y,z} y(i,j,k)}, \quad (9)$$

where $y(i,j,k)_{\text{Resid}}$ represents the value of each pixel (voxel) at (i,j,k) for the residual image $\mathbf{DI}_{\text{Resid}}$ and similarly for image predicted by the HS-CNN algorithm, $\mathbf{DI}_{\text{HS-CNN}}$.

The CNN was trained with stochastic gradient descent momentum optimization. Since one deals with complex images and very large datasets, the gradient descent method does not by itself solve the problem of accurately estimating the optimal values of the weights w . Hence, the algorithm was used with momentum at each iteration, with the momentum defined as the moving average of the gradients; this approach retains the update Δw , the difference between the weights in two consecutive iterations, which is then used in the subsequent update. The momentum was used to prevent the optimization computation from converging to a local minimum or a saddle point. A high momentum parameter accelerates the speed of convergence to the true minimum of the loss function, but setting the momentum parameter at too high a value may give rise to overshooting the true global minimum, thereby making computations unstable. On the other hand, if the momentum coefficient is too low, it cannot reliably help the minimization in order to avoid local minima, thereby slowing down the training. A moving average was also used because the data were just images.

Figure 9 presents a visual comparison of the results, indicating excellent accuracy of the HS-CNN algorithm. The initial LR image represents a very smooth and opaque view of the pores in a shale sample, whereas the enhanced image generated by the HS-CNN reveals features as observed in the reference image. Similarly, the image generated by the BC interpolation method is a smooth

reconstructed image. As expected, the CNN alone, without the large dataset produced by CCSIM algorithms, did not perform well. Note, however, that in the modeling of porous materials, one usually has only a few HR images, which are often insufficient. [92] also made a quantitative comparison using various statistical measures. One measure, for example, was peak signal-to-noise ratio (PSNR) \mathcal{R} , which was used with the images generated by both the bicubic interpolation and the HS-CNN algorithm and compared with that of the reference high-resolution image. \mathcal{R} is defined as follows:

$$\mathcal{R} = 10 \log_{10} \left(\frac{M_I^2}{\sigma^2} \right), \quad (10)$$

where M_I denotes the maximum possible pixel value in the image \mathbf{DI} and σ^2 is the mean-squared error. The computed statistical measures, such that given by Equation 10, indicated the high accuracy of the enhanced images.

4.2 Biological porous media

Every reconstruction method developed for RLPM can also be applied to biological materials, with the results used for training ML algorithms. As an example, Figure 10 presents a 2D MRI image of a baby's brain and three realizations reconstructed from it using the CCSIM algorithm. [101] used a GAN to generate synthetic chest X-rays for infections caused by COVID-19 to train the model. The training data consisted of actual and synthetic chest X-ray images, which were then fed into a customized CNN for deep learning-based chest radiograph classification in order to distinguish the COVID-19 cases from patients with regular pneumonia, along with normal cases free of the disease. The accuracy of the model for detecting COVID-19 cases were 93.94 and 54.55 percent, respectively, with and without data augmentation, i.e., without generating synthetic data by GAN reconstruction, thereby indicating the accuracy of the GAN.

5 Extracting features from an image

To use images of porous media to compute their effective properties, one must first extract the most important features of the images that are at the level of the pixels. Feature extraction is one of the most important and also one of the most challenging problems in image analysis. In addition to extracting the features, one must also interpret them meaningfully. This is, of course, true of any digitized image, but it is true for images of porous materials. Thus, we describe this aspect of the problem.

5.1 Rock-like porous media

We describe the problem using a concrete example and then briefly discuss related works. [102] used images of sandstone to link the effective permeability of the porous media to their morphology. Their proposed network was composed of two ML structures, namely, a CNN for feature extraction and a regular fully connected NN for estimating the permeability. We describe in detail the CNN

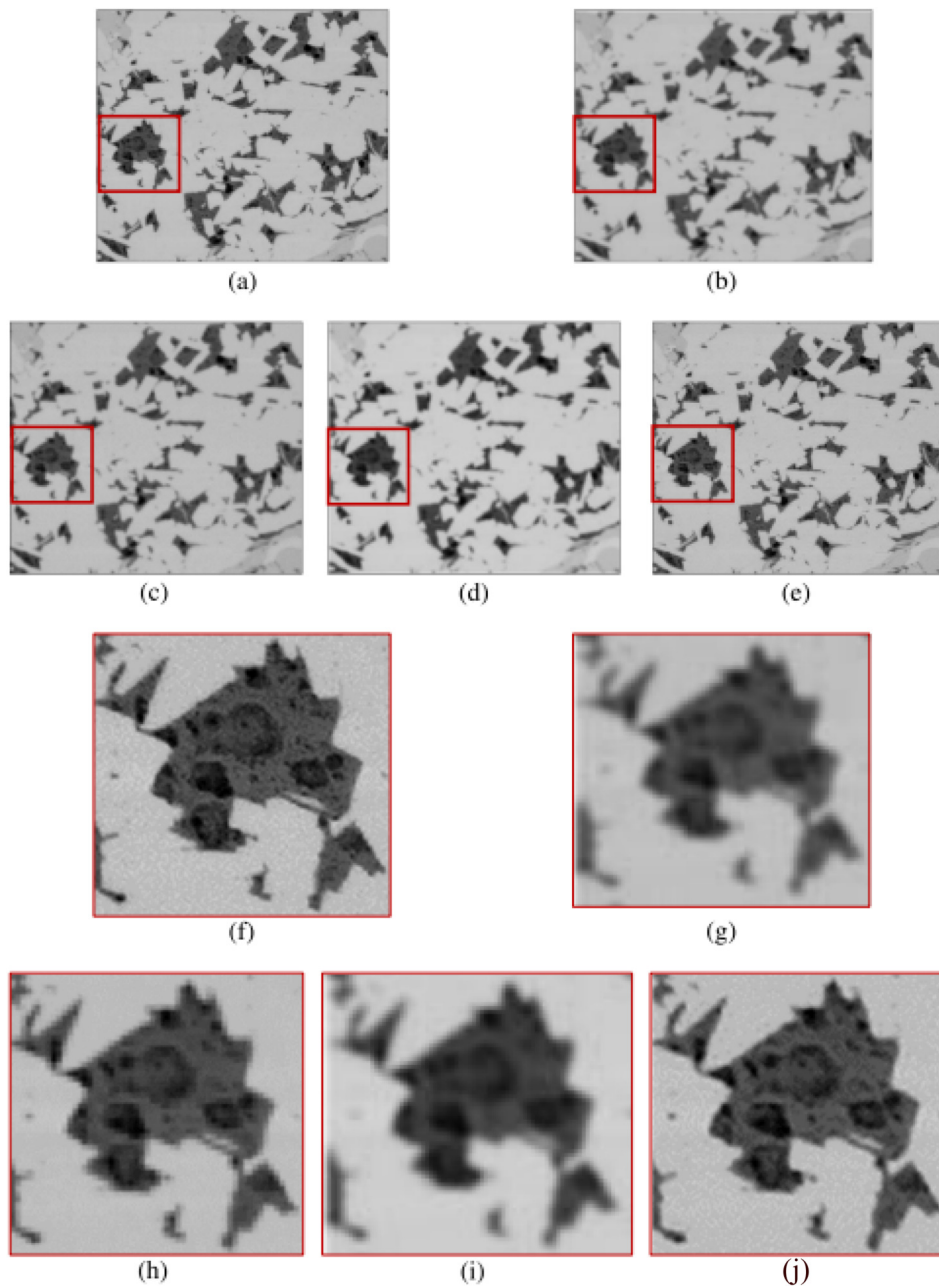


FIGURE 9

Comparison of (a) the reference CT image and (b) the low-resolution CT input, with the image obtained by (c) regular CNN; (d) the bicubic interpolation; and (e) the HS-CNN algorithm. For better comparison, a zoom-in portion is also shown. The zoomed-in portions in each image, corresponding to the top five images in the same order, are also shown for comparison. (f–i) show the zoomed-in part with various resolutions, and compare the accuracy of the methods, with (j) obtained by the HS-CNN method described.

that extracted the features. The filters in the convolution layers slide along the input data, with each producing a specific feature map. The input data, 3D images labeled with their permeabilities, were then convolved using 3D filters, each of size $k \times k \times k$. For simplicity of describing the procedure, the following equation was used, which represents a linear mapping

$$g(x, y, z) = \sum_{i=-m}^m \sum_{j=-m}^m \sum_{k=-m}^m f(x-i, y-j, z-k) w(i, j, k) + b, \quad (11)$$

where f and g represent, respectively, the input and output; (x, y, z) are the coordinates of the voxels in the input; and w and b are the weights and biases, respectively. Each set of the 3D filters produces a 3D output image, which is the input for the next convolutional layer. Note that in Equation 11, only the bias does not depend on the location; otherwise, every pixel value $f(x, y, z)$ is multiplied by a weight, and therefore, the resulting feature map depends on the location. If, however, two pixels have the same value in the input

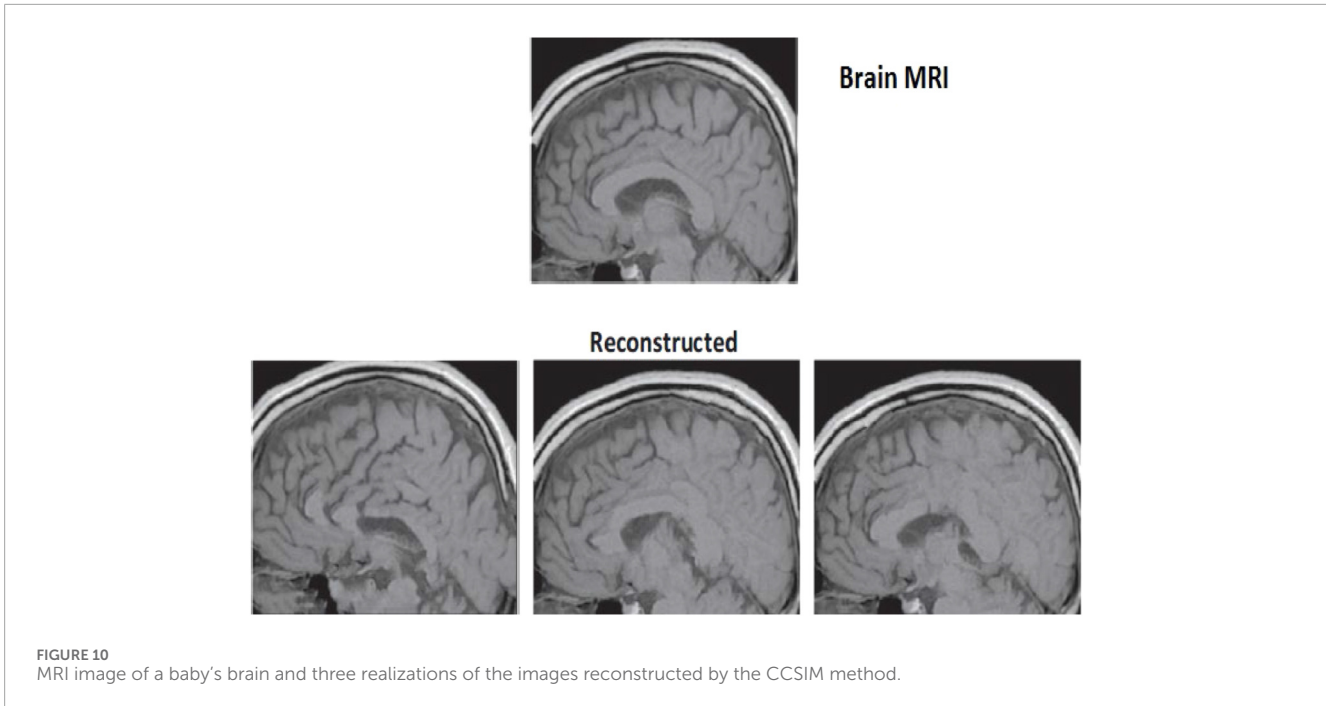


FIGURE 10
MRI image of a baby's brain and three realizations of the images reconstructed by the CCSIM method.

data and the weights are also the same, then the output will also be the same. This problem is addressed by the pooling layer that CNNs usually contain. The convolutional layer convolves the input images using the filters in order to extract their key features. To filter part or all of the input data, they are multiplied by the corresponding matrices, with the results summed up and placed in a single output pixel, as shown in Figure 11, where a 2×2 filter was used. The process is repeated by sliding the filters along the input data using the stride operation in order to generate new output features. Since the filter moves one unit along the data, the stride is one. The entire process of multiplying and summing up represents the convolution operation as it is similar to the classical convolution. In principle, one may require several filters because image recognition requires various feature maps, and multiple filters capture their essential properties. [102] used fifty $5 \times 5 \times 5$ filters. As Figure 11 indicates, the pixels at the center have more influence in the convolution than those on the border of the input matrix (the 3D images), which can be problematic. To remedy the issue and take the effect of the marginal pixels into account, the edges were padded with extra zero-value pixels. The activation function $\mathcal{A}(x)$ used was ReLU, which was applied to the extracted features at each pixel x to produce their maps $\mathcal{A}(x)$. Therefore, several maps were produced that hopefully captured all the important features that connected the input data to the output.

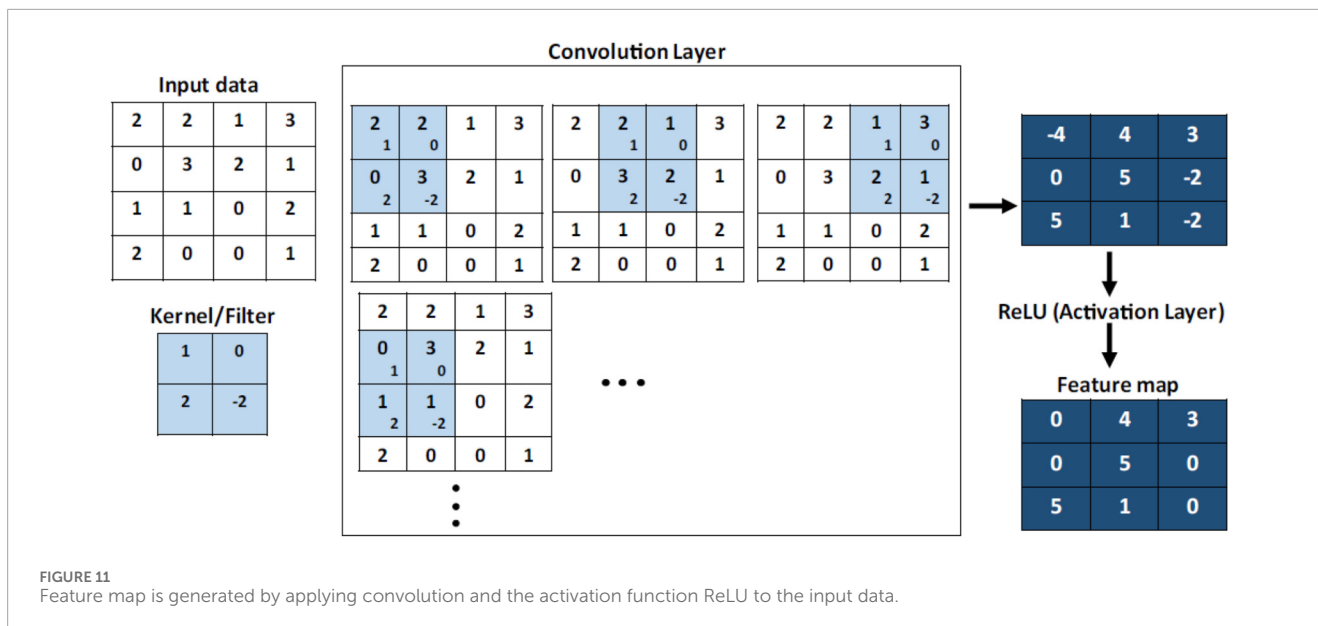
Despite their effectiveness in extracting various features from the input images, the resulting maps may still be limited in terms of recording features' locations in the input data. This means that if the same pattern appears in various locations in the input data due to, for example, cropping, rotation, resolution, and size changes, then its feature map will be different after the convolution layer, thereby giving rise to a major problem with the convolved maps, namely, their sensitivity to the features' location in the input. The reason is that the filters in the convolutional layers are applied with very small strides, usually stride 1, and thus, the maps generated by the

convolutional layer will have very similar locations for each feature in the input. Hence, when a convolutional filter is applied and slid over the image, the output feature map may still depend on the data in each region. To resolve the problem, one may downsample, or coarsen, the feature maps, making them more robust to changes in the features' position using the next layer of the CNN after the activation, the pooling layer, that downsamples the feature maps by summarizing the presence of the features in the map's patches, while preserving the essential features of the data. This not only reduces the size of the data entering the fully connected layer but also reduces the computation. Thus, the pooling layer is very useful when one must preserve a specific feature without considering its exact location in the image because the layer combines all the semantically similar features. Reducing the dimensionality of the feature map also reduces the sensitivity to the location. Note that the pooling operation is specified, rather than learned. The pooling layer acts on each feature map separately in order to generate a new set of the same number of pooled feature maps. The size of the pooling operation or filter is smaller than that of the feature map. Two common functions used in the pooling operation are average pooling, which calculates the average value for each patch on the feature map, and maximum pooling (max-pooling), which computes the maximum value for each patch of the feature map. [98] used the former, which is defined as follows:

$$g(x, y, z) = \sum_{k=0}^{s-1} \sum_{j=0}^{s-1} \sum_{i=0}^{s-1} \left(\frac{1}{s^3} \right) f(sx + i, sy + j, sz + k),$$

where s is the coarsening length scale.

The output of the pooling layers was then flattened to a 1D vector to produce the input to the fully connected layer, the final layer in the proposed network, before the output layer. The fully connected layer keeps a collection of the most important outputs from all the convolutional layers. Since the purpose of the network was regression, the number of neurons in the output layer depended



on the number of targets, i.e., the permeabilities to be predicted. Each neuron in the output layer had a continuous value that represented the prediction for each target. All the feature maps in the last convolutional layer were connected to a unit in a fully connected layer, with the output layer producing the prediction y_i —the permeability for image i —calculated by a logistic regression layer, $y_i = \mathcal{A}_R(w_i v_i + b)$, where $\mathcal{A}_R(x)$ is the activation function of the regression layer, taken to be the logistic function, $[1 + \exp(-x)]^{-1}$.

This basic scheme has been used by many to link various properties of RLPM to their morphology, including diffusivity, elastic moduli, and the dispersion coefficient in solute transport through porous media [103].

5.2 Biological porous media

As mentioned above, extracting feature representations from images at the pixel level is a difficult problem, particularly for MIs. Since CNNs have the capability to learn meaningful features at multiple levels and classify them, their use for feature extraction from medical images has naturally attracted attention. Extracting features from MIs falls within a general class of techniques called content-based image retrieval (CBIR)—also referred to as query by image content (QBIC) and content-based visual information retrieval (CBVIR)—which use computer vision techniques to address the problem of image retrieval, or the search for digital images in large databases. “Content” in this context is a reference to colors, shapes, textures, and similar information that can be extracted from the image, and therefore, “content-based” implies analyzing the contents of the image, rather than such metadata as descriptions associated with the image. A comprehensive review of the application of ML algorithms to the CBIR problem is provided by [104].

[105] combined feature descriptors of CNNs to extract 10^3 features for overlapping patches in 3D MRIs for the prostate and to construct a large feature matrix over all the data volumes. Hashing forests are an approach for mapping a query onto a target in a database, enabling similar nearest neighbors to be searched for

and retrieved efficiently and accurately. This is typically done by learning a hashing function [106] that maps the database entries onto compact binary codes in the Hamming space. A hashing forest is a supervised variant of random forests developed for the task of nearest neighbor retrieval. A Hamming space is the set of all 2^N binary strings or vectors of length N in which two binary strings are considered to be adjacent if they differ only in one position. Then, the distance between any two binary strings, called the Hamming distance, is the number of positions at which the corresponding bits are different. [107] used a CNN with five layers to extract features from the fully connected layers of the CNN. They utilized the last layer and a pre-trained CNN and fed the extracted features to a support vector machine classifier to obtain the distance metric. The conclusion was that incorporating gender information produces better performance than just CNN features.

6 Applications

The use of MIs in the diagnosis of ill patients has a long history. The use of digital images in analyzing problems with various phenomena in RLPM is much more recent. As computational power increased dramatically over the past three decades and new computational approaches for calculating various properties of RLPM, such as the lattice Boltzmann method [108], were developed to simulate various phenomena directly within images, the use of such images has increased significantly. Although the application of simple NNs for regression models correlating various properties of porous media dates back at least 30 years, it was the development of CNNs and other types of NNs and ML algorithms that spurred their widespread use in problems associated with RLPM. An extensive review of ML algorithms to RLPM, as well as problems associated with environmental issues, is provided by [109] and need not be repeated here, while [110] and [95] presented reviews of ML algorithms to MIs. Thus, we describe the applications of the methodologies described above to one particular class of biological

porous media, namely, the human brain, which is the research area of one of authors (A.S.).

6.1 Brain images

The three most common types of brain lesions are gliomas, meningiomas, and pituitary tumors. Thus, we first describe each type of lesion and then discuss the progress that has been made in using ML methods for analyzing and interpreting MIs associated with brain tumors.

According to Mayo Clinic [111], “Glioma is a growth of cells that starts in the brain or spinal cord. The cells in a glioma resemble healthy brain cells called glial cells. Glial cells surround nerve cells and help them function. As a glioma grows, it forms a mass of cells called a tumor. The tumor can grow to press on the brain or spinal cord tissue and cause symptoms.” Gliomas are primary tumors that grow within the central nervous system; a subset of them, glioblastoma (GBM), is the most common primary adult brain tumor. Diagnosis of GBM involves a combination of neurological examination, advanced neuroimaging, such as MRI, and histopathological analysis of composite subtypes after surgical resection or biopsy [112]. The current “gold standard” of treatment for GBM is radiation therapy (RT) with concurrent temozolomide (TMZ) or equivalent alkylating chemotherapy. The former, when added to surgery, improves survival by approximately 6 months [113]. More recent studies have demonstrated that the addition of temozolomide, both concurrently with and following radiation, increases survival by an additional two months [114]. Temozolomide, an FDA-approved chemotherapy drug sold under the brand name Temodar, is used to treat certain types of brain tumors, including GBM and anaplastic astrocytoma. It is an alkylating agent that, by damaging the DNA of cancer cells, slows or stops their growth. Recurrence is, however, inevitable and will almost always occur within the previously radiated area. GBM is highly heterogeneous with multiple cellular subtypes identified within a given tumor on transcriptomic sequencing. Studies have demonstrated that molecular subtypes have varying responsiveness to RT. Moreover, the impact of subtype analysis on radiation delivery is limited as it is usually obtained too late to impact RT planning due to the need to start radiation immediately after surgery. In addition, the high rate of recurrence of GBM suggests that a subset of radio-resistant cells is persistent and that there are likely infiltrative tumor regions not readily apparent on conventional MRI, which, therefore, receive sub-therapeutic doses of radiation.

Given the aforementioned problems—the varying radio-sensitivity, the longitudinal tumor dynamics, and difficulties in identifying recurrent cell reservoirs—there exists a critical gap in the clinical capabilities for providing adaptable, longitudinal care that adequately addresses each person’s unique tumor burden. Current qualitative neuro-imaging approaches provide some distinguishing information that can predict prognostic molecular markers based on such features as texture and density [115], but conventional systems lack the ability to determine the extent of disease infiltration and the underlying molecular heterogeneities to adequately target them based on each patient’s unique tumor burden. Due to such difficulties and the knowledge gap, it was hypothesized that one can bridge the gap by leveraging new MRI technologies to extract relevant imaging features, in addition to patient outcome data, and

construct a predictive NN based on the data. Addressing these limitations, investigators have recently reported on leveraging conventional MRI-guided GBM tissue biopsies to contrast-enhancing and non-enhancing MRI regions in illustrating the potential for ML-based development of predictive radiogenomic modeling of underlying intratumor molecular heterogeneity [116].

A meningioma, according to Mayo Clinic [117], is “a tumor that grows from the membranes that surround the brain and spinal cord, called the meninges. A meningioma is not a brain tumor, but it may press on the nearby brain, nerves, and vessels. Meningioma is the most common type of tumor that forms in the head. Meningiomas occur more often in women. Most meningiomas grow very slowly over many years. They can grow without causing symptoms. However, sometimes, their effects on nearby brain tissue, nerves, or vessels may cause serious disability.” Most meningiomas are not cancerous, but some can be cancerous. More than 39,000 Americans are diagnosed with meningioma every year.

In humans, pituitary tumors, also called pituitary adenomas, are unusual growths that develop in the pituitary gland, or hypophysis, which is an endocrine gland (a network of glands and organs throughout the body) in vertebrates. It is behind the nose at the base of the brain. Some of these tumors cause the pituitary gland to produce excessive amounts of certain hormones that regulate important body functions, while others may cause the gland to produce insufficient amounts of the same hormones. Most pituitary tumors are benign, not cancerous, and typically do not spread to other parts of the body.

Deep NNs have been used extensively for analyzing brain images with multiple applications, including classification of disorders (such as schizophrenia and Alzheimer’s disease); segmentation of tissues, lesions, and tumors; detection and classification of tumors and lesions; predicting survival rate against certain diseases; and image reconstruction. Most ML-based methods learn mappings from local patches to representations to labels. However, if the local patches do not contain the contextual information required for anatomical tasks, then a more refined method may be needed. For example, as mentioned above, [75] used patches obtained through non-uniformly sampling, which was carried out by gradually lowering the sampling rate in patch sides in order to span a larger context. Convolutional NNs have been used extensively in many analyses of brain images, typically obtained by brain MRI, for which there are many public MRI datasets on brain tumor classification, such as Kaggle, OpenNEURO, and fastMRI.

[118] used a three-layer CNN to classify the three types of brain tumors described above. The ReLU activation function was used in the first layer, followed by a structure in which the activation function could be adjusted to normalize the input layer. Dropout operations (i.e., randomly setting the outputs of hidden neurons to zero, usually carried out during the training of an NN) were used in both the second and third layers to mitigate the risk of overfitting. One CNN was used to distinguish between meningioma, glioma, and pituitary tumors with 3,064 enhanced images for 233 patients. The World Health Organization (WHO) classifies the risk levels associated with gliomas into four grades, with grade I being the least dangerous and grade IV being the most dangerous. [118] used a second CNN that differentiated between the three glioma grades of higher risk—grades II, III, and IV—using 516 images for 73 patients. The accuracies of the two distinct cases were, respectively,

96.13 and 98.7 percent. [119] undertook a similar study to classify glioma and meningioma tumors of the brain. They used a CNN for the classification problem and a second one, faster region-CNN (faster R-CNN), for segmentation. Faster R-CNN, proposed by [120], shares full-image convolutional features with the detection NN, which makes it possible to obtain region proposals by essentially cost-free computations. It consists of two modules, with one being a deep CNN that proposes regions and the second one being the faster R-CNN detector [53, 55], which uses the proposed regions. The entire system is a single, unified NN for object detection. [120] used the faster R-CNN with 218 images as the training set, employing a CNN with two convolutional layers. Their model achieved an accuracy of 100 percent for meningioma classification and 87.5 percent for glioma classification, with an average confidence level of 94.6 percent in the segmentation of meningioma tumors.

CNNs were used by [121] (see also [122]) to classify tumor grades (see above), which were segmented using a pre-trained CascadeCNN, after which the deep features were extracted and the tumors were classified. CascadedCNNs were introduced by [123] for the segmentation of brain tumors. Their idea was that since the predictions should, in principle, be influenced by the model's beliefs about the value of nearby labels, one should feed the output probabilities of a first CNN as additional inputs to the layers of a second CNN by concatenating the convolutional layers. Thus, [123] concatenated the output layer of the first CNN with *any* of the layers in the second CNN, which, effectively, represented a cascade of two CNNs with a two-way processing of an image. [121] carried out the classification using a fine-tuned VGG-19 network (see above), a CNN with 16 convolutional layers and 3 fully connected layers. Two datasets, Radiopaedia and brain tumor [124], were employed, which were extended using eight different augmentation techniques. The datasets were divided into 50, 25, and 25 percent sets for, respectively, training, cross-validation, and testing. The accuracy for predicting the Radiopaedia dataset for grade I was, respectively, 90.03 and 95.5 percent without and with data augmentation, with similar accuracies for grades II, III, and IV. The corresponding numbers for the brain tumor dataset were 88.41 and 96.12 percent.

There have also been a few published studies for the specific case of glioblastoma. [125], for example, studied the problem of resection after surgery, considered to be one of the main prognostic factors for patients who have been diagnosed with glioblastoma. To achieve this, one must carry out segmentation and classification of residual tumors from postoperative MR images, and [125] used DCNNs to study the problem. The dataset used consisted of pre- and early post-operative MRI scans from 956 patients (in 12 hospitals in the United States and Europe) who had undergone surgical resection of glioblastoma. With very few exceptions, all the EPMR MRIs (early postoperative magnetic MRIs) were obtained within 72 h after surgery. To have anatomical consistency among the various input sequences, an initial image-to-image registration procedure was implemented. The brain masks were automatically generated using a pre-trained slab-wise AGU-Net model with input of $256 \times 192 \times 32$ voxels. AGU-Net (annotation-guided U-net) was introduced by [126] for rapid one-shot video object segmentation. It consists of two parts, a fully convolutional Siamese NN and a U-Net. In the first part, Siamese NNs are used to label the foreground and background in a new frame that corresponds to a reference object image, while the second part fuses the annotation information

obtained by Siamese networks into the skip connection of a U-net to form an annotation-guided U-net for segmenting a specific target. A Siamese NN uses the same weights while working concurrently on two different input vectors to compute comparable output vectors. Since the AGU-Net analysis required a lower resolution, all inputs were resampled to an isotropic spacing and resized to $128 \times 128 \times 144$ voxels. Based on the analysis of the dataset by the nnU-Net framework (see above), the 3D full-resolution U-Net was selected for further analysis, using $192 \times 128 \times 80$ voxels as the size of the input patch, with the NN using five levels, downsampling using strided convolutional layers (a stride of one was used), and upsampling utilizing transposed convolutional layers. The size of the kernel was $1 \times 3 \times 3$ voxels in the first level and $3 \times 3 \times 3$ for the remaining four, while those of the filter were {32, 64, 128, 256, 320} for the corresponding levels. The loss function was a combination of the Dice score and cross-entropy (see above). All models were initialized using pre-trained weights from the best pre-operative glioblastoma segmentation model. The input layer was the only layer that was adapted to account for the various input combinations. The Adam optimizer was used with an initial learning rate, $l_r = 10^{-3}$. Residual tumor tissue was predicted during inference by each trained model, generating a probability map of the same resolution as the EPMR CE scan. The accuracy of the trained NNs was evaluated based on their ability to segment the residual tumor and classify patients into those with gross total resection and those with residual tumor. Only two classes were considered for segmentation, whereby a positive voxel exhibited tumor tissue, whereas a negative voxel indicated either background or normal tissue. A rest tumor volume threshold was selected for classification that served as the cut-off value. A Dice score of 61 percent was the best achieved segmentation performance, while the best classification performance reached 80 percent. More importantly, the segmentation performance of the most accurate models was comparable to that of human expert raters.

[127] developed an enhanced YOLO (so above) architecture for detecting brain tumors, which improved significantly the accuracy of detection using convolutional block attention modules (CBAMs), squeeze-and-excitation blocks (SE), and residual blocks. The CBAM increases the ability to focus on critical spatial and channel-wise features in MIs, while the SE blocks further improve feature representation by emphasizing important channels. RepVGG blocks [128] can efficiently extract features, while residual blocks help mitigate the vanishing gradient problem. The model was evaluated based on three brain tumor datasets and was demonstrated to provide significant improvements over previous models.

Other medical problems associated with the human brain have also been studied recently using ML approaches. [129] studied the problem of automatic identification of epilepsy in electroencephalography (EEG) signals. They pre-processed the data using adaptive wavelet denoising models in order to remove noise while preserving the actual information-carrying part of the signal. Then, multiple variable permutation entropy and multiple variable multi-scale fuzzy entropy (mvMFE) were extracted from the data that contained complexity and frequency-specific variations. To enhance the discriminative power of the features, uniform manifold approximation and projection were also utilized for non-linear dimensionality reduction. Next, the ResNet neural network (see above), integrated with bi-directional LSTM, was used to capture temporal dynamics and spatial features. When implemented in

Python, the model achieved an accuracy of 94 percent, an F1-score of 96 percent, a recall value of 93 percent, a specificity of 87.70 percent, and a precision of 82.21 percent. [130] used deep CNNs, together with various transfer learning models (see above), to analyze resting state functional MRI (fMRI) images in order to predict the autism disorder features. Their work indicated that VGG16 NN achieves a classification accuracy of 95.8 percent. Their work supports our contention that ML models can be generalized across various neuroimaging-based diagnoses.

7 Summary and conclusion

This study described and discussed recent advances in the applications of machine-learning algorithms, particularly neural networks, to addressing important problems in porous media and materials. The main issue described and discussed in this study is that two important classes of such materials that may seemingly be unrelated have many commonalities. One class consists of what are referred to as rock-like porous media (RLPM), such as soil, concrete, asphalt, and oil and gas reservoirs. A second group consists of BPMs and organs, such as skin, the brain, and lungs. Digital images of BPMs have been used for the diagnosis and treatment of illnesses for several decades, while their utilization in the modeling of various phenomena in RLPM is relatively new, giving rise to what is popularly referred to as “digital rock.” Due to the complexity of such images, along with the need to extract as much information from them as possible, the use of ML approaches, particularly NNs, has been increasing rapidly. We argue that, while the two classes of porous media and materials may appear completely different, they actually have many similarities and that using the images to solve various problems for both classes requires addressing similar issues. As a result, the application of ML algorithms to both types of porous materials is completely similar, even though the goals are seemingly very different. Thus, each scientific community can take advantage of advances in the other to gain a deeper understanding of the problems they study. In particular, we propose that, given that many biological materials and organs are porous materials, the medical community can take advantage of highly advanced approaches for the characterization and modeling of rock-like porous media that have been developed by those who work on such systems as the field of modeling of BPM is replete with *ad hoc* regression models and, in many cases, unrealistic models of BPMs.

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.

References

1. Sahimi M. *Flow and transport in porous media and fractured rock*. 2nd edn. Weinheim: Wiley-VHC (2011).
2. Ghanbarian-Alavijeh B, Hunt AG, Ewing RE, Sahimi M. Tortuosity in porous media: a critical review. *Soil Sci Soc Am J* (2013) 77:1461–77. doi:10.2136/sssaj2012.0435
3. Menon GK, Cleary GM, Lane KE. The structure and function of the stratum corneum. *Int J Pharm* (2012) 435:3–9. doi:10.1016/j.ijpharm.2012.06.005
4. Osseiran S, Dela Cruz J, Jeong S, Wang H, Fthenakis C, Evans CL. Characterizing stratum corneum structure, barrier function, and chemical content of human skin with coherent raman scattering imaging. *Biomed Opt Express* (2018) 26:6425–43. doi:10.1364/BOE.9.006425
5. Liu P, Zhang T, Huang Y. Three-dimensional model of normal human dermal tissue using serial tissue sections. *Front Bioeng Biotechnol* (2024) 12:1347159. doi:10.3389/fbioe.2024.1347159

Author contributions

AS: Data curation, Writing – original draft, Methodology. MS: Conceptualization, Project administration, Data curation, Writing – original draft.

Funding

The author(s) declared that financial support was received for this work and/or its publication. While the work of M.S. for the problems described in this study did not receive any specific funding, his current work on the applications of ML algorithms and NNs to problems in porous media and materials is supported by three National Science Foundation grants. A.S. is supported by a scholarship from New York University Grossman School of Medicine. The grant numbers are CBET 2230593, Hydrol. 2333378.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article 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.

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.

6. Miguel AF. Lungs as a natural porous media: architecture, airflow characteristics and transport of suspended particles. In: J Delgado, editor. *Heat and mass transfer in porous media*. Berlin: Springer (2012). p. 13.
7. DeGroot CT, Straatman AG. Towards a porous media model of the human lung. *AIP Conf Proc* (2012) 1453:69–74. doi:10.1063/1.4711155
8. Hu L, Li Y, Shan X, Zhan H, Pan S, Dou Y, et al. Evaluation of lung function by pore size distribution characteristics of lung tissue based on non-invasive imaging. *J Rad Res Appl Sci* (2023) 16:100666. doi:10.1016/j.jrras.2023.100666
9. Helmberger M, Pienn M, Urschler M, Kullnig P, Stollberger R, Kovacs G, et al. Quantification of tortuosity and fractal dimension of the lung vessels in pulmonary hypertension patients. *PLoS One* (2014) 9:e87515. doi:10.1371/journal.pone.0087515
10. Nicholson C. Sheet and void porous media models for brain interstitial space. *J Roy Soc Interf* (2023) 20:20230223. doi:10.1098/rsif.2023.0223
11. Rusakov DA, Kullmann DM. Geometric and viscous components of the tortuosity of the extracellular space in the brain. *Proc Natl Acad Sci USA* (1998) 95:8975–80. doi:10.1073/pnas.95.15.8975
12. Richesson S, Sahimi M. Flow and transport properties of deforming porous media. I. Permeability. *Transp Porous Media* (2021) 138:577–609. doi:10.1007/s11242-021-01633-y
13. Richesson S, Sahimi M. Flow and transport properties of deforming porous media. II. *Electr Conductivity Transp Porous Media* (2021) 138:611–36. doi:10.1007/s11242-021-01634-x
14. Nicholson C, Phillips JM. Ion diffusion modified by tortuosity and volume fraction in the extracellular microenvironment of the rat cerebellum. *J Physiol* (1981) 321:225–57. doi:10.1113/jphysiol.1981.sp013981
15. Syková E, Nicholson C. Diffusion in brain extracellular space. *Physiol Rev* (2008) 88:1277–340. doi:10.1152/physrev.00027.2007
16. Deen WM. Hindered transport of large molecules in liquid-filled pores. *Aiche J* (1987) 33:1409–25. doi:10.1002/aic.690330902
17. Sahimi M, Jue VL. Diffusion of large molecules in porous media. *Phys Rev Lett* (1989) 62:629–32. doi:10.1103/PhysRevLett.62.629
18. Khademi M, Sahimi M. Static and dynamic properties of supercooled water in small nanotubes. *J Chem Phys* (2016) 145:024502. doi:10.1063/1.4955313
19. Nicholson C, Hrabětová S. Brain extracellular space: the final frontier of neuroscience. *Biophys J* (2017) 113:2133–42. doi:10.1016/j.bpj.2017.06.052
20. Chen KC, Nicholson C. Changes in brain cell shape create residual extracellular space volume and explain tortuosity behavior during osmotic challenge. *Proc Natl Acad Sci USA* (2000) 97:8306–11. doi:10.1073/pnas.150338197
21. Hussain S, Mubeen I, Ullah N, Shah SS, Abduljalil Khan B, Zahoor M, et al. Modern diagnostic imaging technique applications and risk factors in the medical field: a review. *Biomed Res Int* (2022) 6:5164970. doi:10.1155/2022/5164970
22. Vinegar HJ, Wellington SL. Tomographic imaging of three-phase flow experiments. *Rev Sci Instr* (1987) 58:96–107. doi:10.1063/1.1139522
23. Wellington SI, Vinegar HJ. X-ray computerized tomography. *J Pet Technol* (1987) 39:885–98. doi:10.2118/16983-pa
24. Charytanowicz M, Kowalski PA, Lukasik S, Kulczycki P, Czachor H. Deep learning for porous media classification based on micro-CT images. In: *International joint conference on neural networks (IJCNN)*. Padua, Italy (2022). p. 1.
25. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. arXiv:1409.4842 (2015):1–9. doi:10.1109/cvpr.2015.7298594
26. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. arXiv:1608.06993 (2016).
27. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 (2017).
28. Jayachandran ISAJ, Gibbs HC, Laya JC, Qaiser Y, Khan T, Ansari MIMS, et al. An object-based approach to differentiate pores and microfractures in petrographic analysis using explainable, supervised machine learning. *Earth Space Sci* (2024) 11:e2023EA003291. doi:10.1029/2023ea003291
29. Rasaei MR, Sahimi M. Efficient simulation of water flooding in three-dimensional heterogeneous reservoirs using wavelet transformations: application to the SPE-10 model. *Transp. Porous Media* (2008). 72:311–338. doi:10.1007/s11242-007-9152-1
30. Aljamsi A, Sahimi M. Speeding-up image-based simulation of two-phase flow in porous media with lattice-Boltzmann method using three-dimensional curvelet transforms. *Phys Fluids* (2021) 33:113313. doi:10.1063/5.0065857
31. Buades A, Coll B, Morel JM. Non-local means denoising. *Image Process On Line* (2011) 1:208–12. doi:10.5201/1p.2011.bcm_nlm
32. Sahimi M. *Artificial intelligence in science and engineering*. Weinheim: Wiley VCH (2026).
33. Berg S, Saxena N, Shaik M, Pradhan C. Generation of ground truth images to validate micro-CT image-processing pipelines. *The Leading Edge* (2018) 37:412–20. doi:10.1190/tle37060412.1
34. Arganda-Carreras I, Kaynig V, Rueden C, Eliceiri KW, Schindelin J, Cardona A, et al. Trainable weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* (2017) 33:2424–6. doi:10.1093/bioinformatics/btx180
35. Marcellis F, Berg S, Brown B. In: EF Médiçi, AD Otero, editors. *Album of porous media*. Cham: Springer (2023).
36. Marcellis F, Steffen B. North Netherlands gas reservoir. In: EF Médiçi, AD Otero, editors. *Album of porous media*. Cham: Springer (2023).
37. Karimpouli S, Tahmasebi P. Segmentation of digital rock images using deep convolutional autoencoder networks. *Comput Geosci* (2019) 126:142–50. doi:10.1016/j.cageo.2019.02.003
38. Badrinarayanan V, Kendall A, Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* (2017) 39:2481–95. doi:10.1109/TPAMI.2016.2644615
39. Tahmasebi P, Sahimi M. Reconstruction of three-dimensional porous media using a single thin section. *Phys Rev E* (2012) 85:066709. doi:10.1103/PhysRevE.85.066709
40. Tahmasebi P, Sahimi M. Cross-correlation function for accurate reconstruction of heterogeneous media. *Phys Rev Lett* (2013) 110:078002. doi:10.1103/PhysRevLett.110.078002
41. Tahmasebi P, Sahimi M. Enhancing multiple-point geostatistical modeling. 1: graph theory and pattern adjustment. *Water Resour Res* (2016) 52:2074–98. doi:10.1002/2015wr017806
42. Tahmasebi P, Sahimi M. Enhancing multiple-point geostatistical modeling. 2: iterative simulation and multiple distance functions. *Water Resour Res* (2016) 52:2099–122. doi:10.1002/2015wr017807
43. Kim HE, Cosa-Linan A, Santhanam N, Jannesari M, Maros ME, Ganslandt T. Transfer learning for medical image classification: a literature review. *BMC Med Imaging* (2022) 22:69. doi:10.1186/s12880-022-00793-7
44. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *J Am Med Assoc* (2016) 316:2402–10. doi:10.1001/jama.2016.17216
45. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* (2017) 542:115–8. doi:10.1038/nature21056
46. Raghu M, Zhang C, Kleinberg J, Bengio S. Transfusion: understanding transfer learning for medical imaging. arXiv:1902.07208 (2019).
47. Menegola A, Fornaciali M, Pires R, Avila S, Valle E. Towards automated melanoma screening: exploring transfer learning schemes. arxiv:1609.01228 (2016).
48. Setio AAA, Ciompi F, Litjens G, Gerke P, Jacobs C, van Riel S, et al. Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. *IEEE Trans Med Imaging* (2016) 35:1160–9. doi:10.1109/TMI.2016.2536809
49. van Grinsven MJJP, van Ginneken B, Hoyng CB, Theelen T, Sanchez CI. Fast convolutional neural network training using selective data sampling: application to hemorrhage detection in color fundus images. *IEEE Trans Med Imaging* (2016) 35:1273–84. doi:10.1109/TMI.2016.2526689
50. Wolterink JM, Leiner T, de Vos BD, van Hamersvelt RW, Viergever MA, Išgum I. Automatic coronary artery calcium scoring in cardiac CT angiography using paired convolutional neural networks. *Med Image Anal* (2016) 34:123–36. doi:10.1016/j.media.2016.04.004
51. Yang D, Zhang S, Yan Z, Tan C, Li K, Metaxas D. Automated anatomical landmark detection on distal femur surface using convolutional neural network. In: *Proceedings of the IEEE international symposium on biomedical imaging (ISBI)*. Brooklyn, NY (2015). p. 17.
52. de Vos BD, Wolterink JM, de Jong PA, Viergever MA, Išgum I. 2D image classification for 3D anatomy localization: employing deep convolutional neural networks. In: *Proceedings of the SPIE on medical imaging, 9784: image processing* (2016) 97841Y.
53. Kumar A, Sridar P, Quinton A, Kumar RK, Feng D, Nanan R, et al. Plane identification in fetal ultrasound images using saliency maps and convolutional neural networks. In: *2016 IEEE 13th international symposium on biomedical imaging (ISBI)*. Prague, Czech Republic (2016). p. 791.
54. Payer C, Stern D, Bischof H, Urschler M. Regressing heatmaps for multiple landmark localization using CNNs. In: S Ourselin, L Joskowicz, MR Sabuncu, G Unal, W Wells, editors. *Medical image computing and computer-assisted intervention - MICCAI, 9901*. Cham: Springer (2016). p. 230–8. doi:10.1007/978-3-319-46723-8_27
55. Ghesu FC, Georgescu B, Mansi T, Neumann D, Hornegger J, Comaniciu D. An artificial agent for anatomical landmark detection in medical images. In: S Ourselin, L Joskowicz, MR Sabuncu, G Unal, W Wells, editors. *Medical image computing and computer-assisted intervention - MICCAI, 9902*. Cham: Springer (2016). p. 229–37. doi:10.1007/978-3-319-46726-9_27
56. Zheng Y, Liu D, Georgescu B, Nguyen H, Comaniciu D. 3D deep learning for efficient and robust landmark detection in volumetric data. In: N Navab, J Hornegger,

- W Wells, A Frangi, editors. *Proceedings of the medical image computing and computer-assisted intervention*, 9349. Cham: Springer (2015). p. 565–72. doi:10.1007/978-3-319-24553-9_69
57. Baumgartner CF, Kamnitsas K, Matthew J, Smith S, Kainz B, Rueckert D. Real time standard scan plane detection and localisation in fetal ultrasound using fully convolutional neural networks. In: S Ourselin, L Joskowicz, MR Sabuncu, G Unal, W Wells, editors. *Medical image computing and computer-assisted intervention - MICCAI*, 9901. Cham: Springer (2016). p. 203–11. doi:10.1007/978-3-319-46723-8_24
58. Küstner T, Fuin N, Hammernik K, Bustin A, Qi H, Hajhosseiny R, et al. CINENet: deep learning-based 3D cardiac CINE MRI reconstruction with multi-coil complex-valued 4D spatio-temporal convolutions. *Sci Rep* (2020) 10:article 13710. doi:10.1038/s41598-020-70551-8
59. Chen H, Dou Q, Ni D, Cheng JZ, Qin J, Li S, et al. Automatic fetal ultrasound standard plane detection using knowledge transferred recurrent neural networks. In: N Navab, J Hornegger, W Wells, A Frangi, editors. *Proceedings of the medical image computing and computer-assisted intervention*, 9349. Cham: Springer (2015). p. 507–14. doi:10.1007/978-3-319-24553-9_62
60. Kong B, Zhan Y, Shin M, Denny T, Zhang S. Recognizing end-diastole and end-systole frames via deep temporal regression network. In: S Ourselin, L Joskowicz, MR Sabuncu, G Unal, W Wells, editors. *Medical image computing and computer-assisted intervention - MICCAI*, 9901. Cham: Springer (2016). p. 264–72. doi:10.1007/978-3-319-46726-9_31
61. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. arXiv:1506.02640 (2015).
62. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. arXiv:1612.08242 (2016).
63. Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv:1804.02767 (2018).
64. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934 (2020).
65. Liu J, Wang D, Wei Z, Lu L, Kim L, Turkbey E, et al. Colitis detection on computed tomography using regional convolutional neural networks. *Med Phys* (2017) 44:4630–42. doi:10.1002/mp.12399
66. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 (2013).
67. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2015).
68. Girshick R Fast R-CNN, arXiv:1504.08083 (2014).
69. He K, Zhang X, Ren S, Sun J. *Spatial pyramid pooling in deep convolutional networks for visual recognition* (2014). arXiv:1406.4729.
70. Cicek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: S Ourselin, L Joskowicz, MR Sabuncu, G Unal, W Wells, editors. *Medical image computing and computer-assisted intervention - MICCAI*, 9901. Cham: Springer (2016). p. 424–32. doi:10.1007/978-3-319-46723-8_49
71. Drozdal M, Vorontsov E, Chartrand G, Kadoury S, Pal C. The importance of skip connections in biomedical image segmentation. arXiv:1608 (2016):04117. doi:10.48550/arXiv.1608.04117
72. Isensee F, Jaeger PF, Kohl SAA, Petersen J, Maier-Hein KM. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat Methods* (2021) 18:203–11. doi:10.1038/s41592-020-01008-z
73. Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: the missing ingredient for fast stylization. arXiv:1607.08022 (2016).
74. Kamnitsas K, Ledig C, Newcombe VF, Simpson JP, Kane AD, Menon DK, et al. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med Image Anal* (2017) 36:61–78. doi:10.1016/j.media.2016.10.004
75. Ghafoorian M, Karssemeijer N, Heskes T, van Uden IWM, de Leeuw FE, Marchiori E, et al. Non-uniform patch sampling with deep convolutional neural networks for white matter hyperintensity segmentation. In: *2016 IEEE 13th international symposium on biomedical imaging (ISBI)*. Prague, Czech Republic (2016). p. 1414.
76. Park SC, Park MK, Kang MG. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process. Mag* (2003) 20:21. doi:10.1109/MSP.2003.1203207
77. Glasner D, Bagon S, Irani M. Super-resolution from a single image. In: *Proceedings of IEEE 12th international conference on computer vision*. Kyoto, Japan (2009). p. 349.
78. Sahimi M, Tahmasebi P. Reconstruction, optimization, and design of heterogeneous materials and media: basic principles, computational algorithms, and applications. *Phys Rep* (2021) 939:1–82. doi:10.1016/j.physrep.2021.09.003
79. Kim J, Lee JK, Lee KM. Deeply-recursive convolutional network for image superresolution. arXiv:1511 (2015):04491. doi:10.48550/arXiv.1511.04491
80. Mao X, Shen C, Yang YB. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Adv Neural Infor Process. Syst* (2016) 29:2802. doi:10.48550/arXiv.1603.09056
81. Dong C, Loy CC, He K, Tang X. Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell* (2014) 38:295–307. doi:10.1109/TPAMI.2015.2439281
82. Dong C, Loy CC, Tang X. Accelerating the super-resolution convolutional neural network. In: B Leibe, J Matas, N Sebe, M Welling, editors. *Computer vision - ECCV 2016*, 9906. New York: Springer (2016). p. 367.
83. Odena A, Dumoulin V, Olah C. Deconvolution and checkerboard artifacts. *Distill* (2015) 10:23915.
84. Shi W, Caballero J, Huszar F, Totz J, Aitken AP, Bishop R, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. arXiv:1609 (2016):05158. doi:10.48550/arXiv.1609.05158
85. Wang YD, Armstrong RT, Mostaghimi P. Boosting resolution and recovering texture of 2D and 3D micro-CT images with deep learning. *Water Resour Res* (2020) 56:e2019WR026052. doi:10.1029/2019wr026052
86. Wang YD, Mostaghimi P, Armstrong RT. A super resolution dataset of digital rocks (DRSRD1): sandstone and carbonate. arXiv (2019). doi:10.17612/P7D38H
87. Lim B, Son S, Kim H, Nah S, Lee KM. Enhanced deep residual networks for single image super-resolution. In: *Proceedings of 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, Honolulu, USA (2017). p. 1132.
88. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of 2017 IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, USA (2017). p. 105.
89. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. arXiv:1502.01852 (2015):1026–34. doi:10.1109/iccv.2015.123
90. Jackson SJ, Niu Y, Manoorkar S, Mostaghimi P, Armstrong RT. Deep learning of multiresolution X-ray micro-computed-tomography images for multiscale modeling. *Phys Rev Appl* (2022) 17:054046. doi:10.1103/physrevapplied.17.054046
91. Oktay O, Bai W, Lee M, Guerrero R, Kamnitsas K, Caballero J, et al. Multi-input cardiac image super-resolution using convolutional neural networks. In: S Ourselin, L Joskowicz, MR Sabuncu, G Unal, W Wells, editors. *Medical image computing and computer-assisted intervention - MICCAI*, Vol. 9902. Cham: Springer (2016). p. 246–54. doi:10.1007/978-3-319-46726-9_29
92. Kamrava S, Tahmasebi P, Sahimi M. Enhancing images of shale formations by a hybrid stochastic and deep learning algorithm. *Neural Netw* (2019) 118:310–20. doi:10.1016/j.neunet.2019.07.009
93. Golkov V, Dosovitskiy A, Sperl J, Menzel M, Czisch M, Samann P, et al. Q-space deep learning: twelve-fold shorter and model-free diffusion MRI scans. *IEEE Trans Med Imaging* (2016) 35:1344–51. doi:10.1109/TMI.2016.2551324
94. Li M, Jiang Y, Zhang Y, Zhu H. Medical image analysis using deep learning algorithms. *Front Public Health* (2023) 11:1273253. doi:10.3389/fpubh.2023.1273253
95. Mall PK, Singh PK, Srivastav S, Narayan V, Paprzycki M, Jaworska T, et al. A comprehensive review of deep neural networks for medical image processing: recent developments and future opportunities. *Health Analytic* (2023) 4:100216. doi:10.1016/j.health.2023.100216
96. Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge: MIT Press (2016).
97. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. In: Z Ghahramani, M Welling, C Cortes, ND Lawrence, KQ Weinberger, editors. *Advances in neural information processing systems*, 27. New York: Curran Association (2014). p. 2672.
98. Mosser L, Dubrulle O, Blunt MJ. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Phys Rev E* (2017) 96:043309. doi:10.1103/PhysRevE.96.043309
99. Mosser L, Dubrulle O, Blunt MJ. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Math Geosci* (2020) 52:53–79. doi:10.1007/s11004-019-09832-6
100. Iassonov P, Gebrenegus T, Tuller M. Segmentation of X-ray computed tomography images of porous materials: a crucial step for characterization and quantitative analysis of pore structures. *Water Resour Res* (2009) 45:W09415. doi:10.1029/2009wr008087
101. Sakib S, Tazrin T, Fouda MM, Fadlullah ZM, Guizani M. DL-CRC: deep learning-based chest radiograph classification for COVID-19 detection: a novel approach. *IEEE Access* (2020) 8:171575–89. doi:10.1109/ACCESS.2020.3025010
102. Kamrava S, Tahmasebi P, Sahimi M. Linking morphology of porous media to their macroscopic permeability by deep learning. *Transp Porous Media* (2020) 131:427–48. doi:10.1007/s11242-019-01352-5
103. Kamrava S, Im J, de Barros FJP, Sahimi M. Estimating dispersion coefficient in flow through heterogeneous porous media by a deep convolutional neural network. *Geophys Res Lett* (2021) 48:e2021GL094443. doi:10.1029/2021gl094443
104. Dubey SR. A decade survey of content based image retrieval using deep learning. arXiv:2012.00641v2 (2020).

105. Shah A, Conjeti S, Navab N, Katouzian A. Deeply learnt hashing forests for content based image retrieval in prostate MR images. In: *Proceedings of the SPIE 9784* (2016). doi:10.1016/j.media.2016.05.010
106. Conjeti S, Katouzian A, Kazi A, Mesbah S, Beymer D, Syeda-Mahmood TF, et al. Metric hashing forests. *Med Image Anal* (2016) 34:13–29. doi:10.1016/j.media.2016.05.010
107. Anavi Y, Kogan I, Gelbart E, Geva O, Greenspan H. A comparative study for chest radiograph image retrieval using binary texture and deep learning classification. In: *Proceedings of 37th annual international conference of the IEEE engineering in medicine and biology society*. Milan, Italy: EMBC (2015). p. 2940.
108. Chen S, Doolen GD. Lattice boltzmann method for fluid flows. *Annu Rev Fluid Mech* (1998) 30:329–64. doi:10.1146/annurev.fluid.30.1.329
109. Tahmasebi P, Kamrava S, Bai T, Sahimi M. Machine learning in geo- and environmental sciences: from small to large scale. *Adv Water Resour* (2020) 142:103619. doi:10.1016/j.advwatres.2020.103619
110. Litjens G, Kooi T, Ehteshami Bejnordi B, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. *Med Image Anal* (2017) 42:60–88. doi:10.1016/j.media.2017.07.005
111. Mayo Clinic. Glioma (2025). Available online at: <https://www.mayoclinic.org/diseases-conditions/glioma/symptoms-causes/syc-20350251> (Accessed August 15, 2025).
112. Wirsching HG, Galanis E, Weller M. Glioblastoma. *Handb Clin Neurol* (2016) 134:381–97. doi:10.1016/B978-0-12-802997-8.00023-2
113. Walker MD, Green SB, Byar DP, Alexander E, Batzdorf U, Brooks WH, et al. Randomized comparisons of radiotherapy and nitrosoureas for the treatment of malignant glioma after surgery. *N Engl J Med* (1980) 303:1323–9. doi:10.1056/NEJM198012043032303
114. Stupp R, Taillibert S, Kanner AA, Kesari S, Steinberg DM, Toms SA, et al. Maintenance therapy with tumor-treating fields plus temozolomide vs temozolomide alone for glioblastoma: a randomized clinical trial. *JAMA* (2015) 31:2535–43. doi:10.1001/jama.2015.16669
115. Kickingereder P, Neuberger U, Bonekamp D, Piechotta PL, Götz M, Wick A, et al. Radiomic subtyping improves disease stratification beyond key molecular, clinical, and standard imaging characteristics in patients with glioblastoma. *Neuro-Oncol* (2018) 18:848–57. doi:10.1093/neuonc/nox188
116. Hu LS, D'Angelo F, Weiskittel TM, Caruso FP, Fortin Ensign SP, Blomquist MR, et al. Integrated molecular and multiparametric MRI mapping of high-grade glioma identifies regional biologic signatures. *Nat Commun* (2023) 28:6066. doi:10.1038/s41467-023-41559-1
117. Mayo Clinic. Meningioma (2025). Available online at: <https://www.mayoclinic.org/diseases-conditions/meningioma/symptoms-causes/syc-20355643> (Accessed August 15, 2025).
118. Sultan HH, Salem NM, Al-Atabany W. Multi-classification of brain tumor images using deep neural network. *IEEE Access* (2019) 7:69215–25. doi:10.1109/access.2019.2919122
119. Kaldera HNTK, Gunasekara SR, Dissanayake MB. Brain tumor classification and segmentation using faster R-CNN. In: *Proceedings of 2019 advances in science and engineering technology international conferences (ASET)*. Dubai, United Arab Emirates (2019). p. 1.
120. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. arXiv:1506.01497v3 (2016).
121. Sajjad M, Khan S, Muhammad K, Wu W, Ullah A, Baik SW. Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *J Comput Sci* (2019) 30:174–82. doi:10.1016/j.jocs.2018.12.003
122. Swati ZNK, Zhao Q, Kabir M, Ali F, Ali Z, Ahmed S, et al. Brain tumor classification for MR images using transfer learning and fine-tuning. *Comput Med Imaging Graph* (2019) 75:34–46. doi:10.1016/j.compmedimag.2019.05.001
123. Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, et al. Brain tumor segmentation with deep neural networks. *Med Image Anal* (2017) 35:18–31. doi:10.1016/j.media.2016.05.004
124. Cheng J, Huang W, Cao S, Yang R, Yang W, Yun Z, et al. Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS One* (2015) 10:e0140381. doi:10.1371/journal.pone.0140381
125. Helland RH, Ferles A, Pedersen A, Kommers I, Ardon H, Barkhof F, et al. Segmentation of glioblastomas in early post-operative multi-modal MRI with deep neural networks. *Sci Rep* (2023) 13:18897. doi:10.1038/s41598-023-45456-x
126. Yin Y, Xu D, Wang X, Zhang L. AGUnet: annotation-guided U-net for fast one-shot video object segmentation. *Pattern Recognit* (2021) 110:107580. doi:10.1016/j.patcog.2020.107580
127. Khan F, Katsagannis G, Sengar SS. RepSE-CBAMNet: a hybrid attention-enhanced CNN for brain tumor detection. *Stud Health Technol Inform* (2025) 327:567–71. doi:10.3233/SHTI250401
128. Balakrishnan T, Sengar SS. RepVGG-GELAN: enhanced GELAN with VGG-STYLE ConvNets for brain tumour detection. arXiv:2405.03541 (2024).
129. Sikarwar SS, Rana AK, Sengar SS. Entropy-driven deep learning framework for epilepsy detection using electro encephalogram signals. *Neurosci* (2025) 577:12. doi:10.1016/j.neuroscience.2025.05.003
130. Lakshmi PB, Reddy VD, Ghosh S, Sengar SS. Classification of autism spectrum disorder based on brain image data using deep neural networks. In: V Bhateja, XS Yang, MC Ferreira, SS Sengar, CM Travieso-Gonzalez, editors. *Evolution in computational intelligence. FICTA 2023. Smart innovation, systems and technologies*. Singapore: Springer (2023).