Neurocomputing 616 (2025) 128898

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# RockNet: Deep progressive lithology recognition model based on feature saliency and fusion

Xiangyuan Zhu [a], Mincan Li [b,*], Zhiming Lan [a], Jianguo Chen [c], Zerui Li [a], Keqin Li [b,d]

[a] School of Computer Science and Software, Zhaoqing University, Zhaoqing 516061, China
[b] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[c] School of Software Engineering, Sun Yat-sen University, Zhuhai 519082, China
[d] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Communicated by N. Zeng

Dataset link: https://github.com/ZQU-BD/RockNet

*Keywords:*
Deep progressive learning
Lithology recognition
Local feature salience
Multi-channel feature fusion

ABSTRACT

Accurate lithology recognition is pivotal for comprehending subsurface structures and forecasting resource reservoirs in geological exploration. Most existing approaches rarely utilize multi-view heterogeneous rock microscopic images, limiting recognition performance in modern geological practices. To tackle the challenges above, we propose a deep progressive lithology recognition model named RockNet for rock microscopic images, based on local feature saliency and feature fusion. RockNet includes Multi-channel Feature Fusion (MFF) blocks and Local Feature Salience (LFS) blocks. The MFF block captures and fuses hierarchical features from each view of rock images, while the LFS block extracts subtle information across different views. In addition, we further design a novel loss function and a multi-scale prediction fusion strategy to optimize the training and inference process. Finally, RockNet adopts a deep progressive learning strategy to enhance its ability to recognize complex lithological patterns. Experimental results show that RockNet outperforms 12 comparative methods regarding accuracy, precision, recall, F1 score, and specificity. Our work will assist oil and gas exploration and groundwater resources assessment, contributing significantly to resource development and sustainable environmental stewardship.

## 1. Introduction

Within Earth Science, lithology recognition holds immense importance in comprehending subsurface conditions and geological formations [1–3]. Typically, rock photomicrographs constitute heterogeneous multi-view data, each containing numerous instances. The lithology recognition process entails categorizing and describing rock types based on visual and quantifiable characteristics, including mineral composition, texture, color, and structure. This recognition serves as a cornerstone for numerous applications, including petroleum exploration, groundwater resource assessment, environmental studies, and construction projects. Accurate and efficient lithology recognition offers valuable insights into the properties and dynamics of geological formations, facilitating informed decision-making in hazard assessment and engineering initiatives.

Various methodologies have been employed in lithology recognition, including thin section analysis [4], well logging [5], remote sensing [6], and laboratory testing [7]. Among these methods, rock thin section analysis is widely utilized because it can faithfully encode information that is not visible to the naked eye, offering high confidence in results. To ensure repeatable observations of rock thin sections, rock microscopic images are captured and saved as digital images under cross-polarized light (XPL) and plane-polarized light (PPL) using a petrographic microscope, as shown in Fig. 1.

Under PPL, observers can discern the rock's overall color, grain size, shape, and mineral arrangement. Rotating the thin section between crossed polarizers reveals how minerals interact with polarized light, providing insights into mineral relationships and rock fabric. Consequently, images captured under PPL and XPL of the same thin section exhibit heterogeneity, showcasing various features and lacking uniformity or consistency in visual appearance. In addition, capturing all the detailed information in a thin section with a single view is impractical. Therefore, it is recommended to capture multiple PPL and XPL images from various viewpoints, adhering to a specific magnification scale for a comprehensive structural view.

Convolutional Neural Networks (CNNs) [8–10] are employed to classify lithology based on well-log images or thin sections. Some feature fusion algorithms, such as concatenation, attention mechanisms, or

* Corresponding author.
*E-mail addresses:* zxycs@zqu.edu.cn (X. Zhu), limc@hnu.edu.cn (M. Li), lzm_mm2000@163.com (Z. Lan), chenjg33@mail.sysu.edu.cn (J. Chen), 13172512613@163.com (Z. Li), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.neucom.2024.128898
Received 13 August 2024; Received in revised form 29 September 2024; Accepted 6 November 2024
Available online 15 November 2024

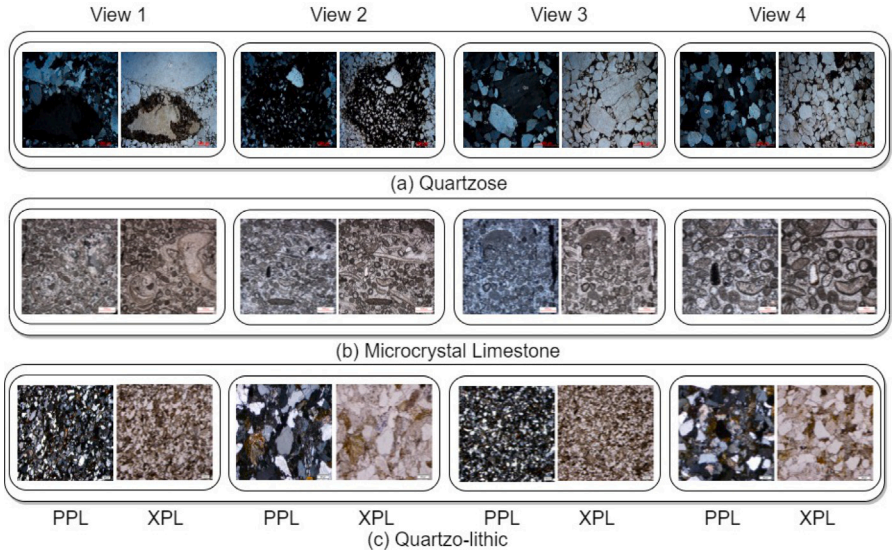



**Fig. 1.** PPL and XPL image examples: quartzose, microcrystalline Limestone, and quartzo-Lithic rocks from four views.

ensemble methods, are designed to combine information from multiple sources or modalities. While extensive research has been conducted on lithology recognition, few studies consider all the features and heterogeneity of multi-view rock images, which limits the classification performance in modern geological practices. First, heterogeneous PPL and XPL images exhibit inconsistent yet complementary characteristics such as color, interference patterns, and extinction angle. In addition, multi-view images introduce challenges such as position shifting, scale variations, and information redundancy. Moreover, microscopic image data in practical scenarios are often insufficient, and categories may be relatively imbalanced.

In this paper, to tackle the challenges above, we propose a CNN-based deep progressive learning model named RockNet to identify lithology from multi-view heterogeneous rock microscopic images. RockNet aims to address the complexities of lithology recognition by leveraging a progressive learning strategy while mitigating the limitations associated with data heterogeneity and category imbalances. The multi-channel feature fusion and progressive learning mechanisms in RockNet significantly enhance the accuracy and reliability of the lithology identification procedure. Our contributions are four-fold:

- A new category-aware augmentation method is designed to alleviate the impact of category imbalance on model training. This method integrates key strategies including minority oversampling, category-balanced mini-batch generator, global and local feature augmentation, and automatic data enhancement.
- A CNN-based deep progressive learning model is proposed for exploring multi-view heterogeneous rock microscopic images. An MFF block is designed to capture and fuse hierarchical features from each view, while an LFS block is presented to integrate and highlight subtle information across different views.
- A progressive learning mechanism is presented to ensure the model refines its comprehension of lithological features over successive stages. This mechanism effectively improves the overall accuracy and reliability of the lithology identification procedure.
- A novel loss function and a multi-scale testing and prediction fusion strategy are developed to optimize the training and inference process through a pre-determined loss contribution factor.

The rest of the paper is organized as follows. Section 2 provides an overview of existing research on lithology recognition, multi-view feature learning, and progressive learning. Section 3 outlines the foundation and problem definition. Section 4 introduces the architecture and core modules of the proposed RockNet algorithm. Section 5 assesses the comparison experiments. Finally, Section 6 summarizes the paper.

## 2. Related work

### 2.1. Lithology recognition methods

Advancements in CNNs have revolutionized traditional lithology recognition, effectively addressing the challenges of long analysis cycles, heavy labor intensity, and the need for high levels of expertise. These advancements offer new tools and techniques for rapidly and precisely categorizing rock types [11]. In [12], CNN-based models were proposed for carbonate petrography identification. Ma et al. [13] developed an improved squeeze-and-excitation model to hierarchically classify rock thin sections. Their approach first categorizes the dataset into three main groups–sedimentary, metamorphic, and igneous rocks–and further subdivides them into 105 second-level rock categories. Dawson et al. [14] evaluated the performance of the nine CNN architectures on transfer learning for carbonate core identification. The results indicated that Inception-V3 performs remarkably well on medium to large datasets, while VGG19 achieves competitive performance on smaller datasets. However, most existing work employ single-view feature learning for either rock image classification [15–17] or rock thin section recognition [18].

### 2.2. Multi-view feature learning

Multi-view feature learning is commonly employed in various applications due to its superior model accuracy compared to single-view feature learning methods. Chaganti et al. [19] introduced a feature fusion method based on multiple views for malware identification. This method gathers static and dynamic features and fuses all the selected features to differentiate malware executable files. Jia et al. [20] designed a network to learn shared features across different views and specific features from each view while reducing feature redundancy. Zeng et al. [21] described a dual-pathway multiscale network for detecting image forgeries. This network aligns visual features with edge information and employs variational convolutions and multiscale fusion to achieve robust region localization. In [22], a learning strategy based on multiple views was presented. This strategy creates multi-resolution tumor-centered image groups and applies a homogeneous bilinear network in each view. As noted, rock photomicrographs are

<table>
<tr><td colspan="6">Allochthonous Limestones</td><td colspan="3">Autochthonous Limestones</td></tr>
<tr><td colspan="6">Original components not organically bound during deposition</td><td colspan="3">Original components organically bound during deposition</td></tr>
<tr><td colspan="4">Less than 10% > 2 mm components</td><td colspan="2">Greater than 10% > 2mm components</td><td rowspan="3">By organisms which act as baffles</td><td rowspan="3">By organisms which encrust and bind</td><td rowspan="3">By organisms which build a rigid framework</td></tr>
<tr><td colspan="3">Contains lime mud (<0.03mm)</td><td>No lime mud</td><td rowspan="2">Matrix supported</td><td rowspan="2">>2mm component support</td></tr>
<tr><td colspan="2">Mud-supported</td><td colspan="2" rowspan="2">Grain-supported</td></tr>
<tr><td>Less than 10% grains</td><td>Greater than 10% grains</td><td>Mudstone</td><td>Wackestone</td><td>Packstone</td><td>Grainstone</td><td>Floatstone</td><td>Rudstone</td><td>Bafflestone</td><td>Bindstone</td><td>Framestone</td></tr>
</table>

**Fig. 2.** Limestone recognition criteria used in this study.

heterogeneous multi-view data, with each view presenting a unique semantic perspective of the rock sample. However, most existing approaches employ Siamese networks or one-view-one-network strategies to extract and fuse multi-view features, which consumes substantial computing resources.

### 2.3. Progressive learning

Progressive learning, which mimics the human learning process, is an effective approach in various real-world applications due to its ability to retain previously acquired knowledge while integrating new information. Du et al. [23] incorporated a progressive training strategy with feature fusion at different granularities. However, their method is tailored for fine-grained visual classification, which may limit its applicability to other types of visual tasks or domains. Huang et al. [24] developed a progressive training strategy named PLFace, utilizing a new progressive learning loss for deep face recognition. During various training phases, PLFace fine-tunes the weights of samples with and without masks. However, this progressive tuning of weights may introduce additional computational overhead, making the training process slower. In [25], Song et al. proposed an approach to denoise medical perfusion imaging using a progressive training strategy. The model is trained jointly to predict more accurate noise, enhancing network performance. Hu et al. [26] introduced a new progressive learning model called $\ell$-DARTS, which enhances the original DARTS model by reducing its depth for faster searches and by introducing a channel fusion compensation module to maintain accuracy. Additionally, it employs an enhanced regularization technique to balance operation preferences. While progressive learning offers significant advantages in retaining and integrating knowledge, its application is often domain-specific and comes with several limitations related to scalability, complexity, and computational demands.

## 3. Foundation and problem definition

### 3.1. Rock classification and naming scheme

Limestone and sandstone are two prominent sedimentary rocks. However, the academic community lacks a standardized and definitive system for categorizing and naming sedimentary rocks [27]. The main challenges are the diverse material composition, varied occurrences and textures, strong heterogeneity, and small grain sizes that are difficult to observe. In this work, we apply the modified recognition rules proposed by Embry and Klovan [28]. The recognition rules are primarily based on the content of grains and cement blocks in the rock samples, grain types, support methods, and how the original components are organically bound. Fig. 2 shows the recognition criteria for limestone.

Following the nomenclature scheme introduced by Garzanti [29], sandstones are classified according to the relative abundance of their three main components (lithic fragments (L), feldspars (F), and quartz (Q)). Suppose one main component does not exceed 10%. In that case, the sandstones are categorized into six types: feldspatho-lithic (FL), litho-feldspathic (LF), litho-quartzose (LQ), quartzo-lithic (QL), quartzo-feldspathic (QF), and feldspatho-quartzose (FQ). These correspond to the six trapezoidal fields within the QFL triangle, as illustrated in Fig. 3. Additionally, the three rhombus fields at the triangle's vertices, where two of the three main components are less than 10%, are labeled simply as Q for quartzose, L for lithic, and F for feldspathic.

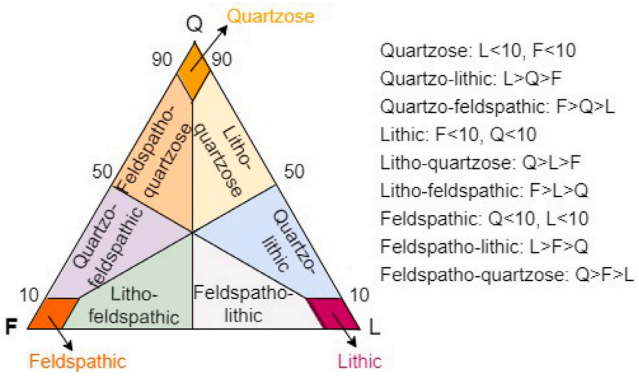



**Fig. 3.** Simplified sandstone classification system with percentage indicators.

### 3.2. Problem definition

Given a dataset $\mathcal{D}$ of labeled rock thin section micrograph, as defined in Eq. (1). It consists of three subsets: a test set $\mathcal{D}^{\text{te}}$, a validation set $\mathcal{D}^{\text{va}}$, and a training set $\mathcal{D}^{\text{tr}}$. The sizes of these subsets are denoted by $N_{\text{te}}$, $N_{\text{va}}$, and $N_{\text{tr}}$, respectively.

$$\begin{cases} \mathcal{D}^{\text{tr}} = \{x_i^{\text{tr}}, y_i^{\text{tr}}\}_{i=1}^{N_{\text{tr}}} \\ \mathcal{D}^{\text{va}} = \{x_i^{\text{va}}, y_i^{\text{va}}\}_{i=1}^{N_{\text{va}}} \\ \mathcal{D}^{\text{te}} = \{x_i^{\text{te}}, y_i^{\text{te}}\}_{i=1}^{N_{\text{te}}} \end{cases} \tag{1}$$

where $x_i$ is a rock image, $y_i$ is the classification label of $x_i$, while te, va, and tr denote the test set, validation set, and training set, respectively.

We build a Deep CNN (DCNN) network with $l$ hidden layers for feature extraction. We divide the DCNN into a feature extractor $f$ with weights $W$ and a linear classifier $h$ with weights $\theta$. To extract features, we introduce two-dimensional convolution (Conv2D), batch normalization (BN), and Sigmoid linear unit (SiLU) activation function layer by layer to the input image $x_i$. Then, the corresponding output logits of the network are obtained. The network predictions $\hat{y}_i$ are achieved by utilizing the softmax function for the logits $X_i^{(l)}$ of the last

hidden layer $l$, as defined as:

$$\begin{cases} X_i^{(j)} = \mathrm{SiLU}(\mathrm{BN}(f^{(j)}(X_i^{(j-1)}; W^{(k)}))) \\ \quad = \mathrm{SiLU}(\mathrm{BN}(\mathrm{Conv2D}(c, dim, r, s, p))) \\ \hat{y}_i = \mathrm{softmax}(X_i^{(l)}; \theta) \end{cases} \tag{2}$$

where $X_i^{(0)} = x_i$, $f^{(j)}(\cdot)$ denotes the convolutional operation with weights $W^{(k)}$, and $j$ and $k$ represent the indices of hidden layers and convolutional operations, respectively. The Conv2D operation is parameterized by the input channels $c$, the number of filters $dim$, filter size $r$, stride $s$, and padding $p$.

*Rock Lithology Recognition*. Given the dataset $D$, we aim to learn a classifier $h_{W,\theta} : x_i \to \hat{y}_i$, which is parameterized by $W$ and $\theta$. The classifier maps each image $x_i$ to its predicted category $\hat{y}_i$. The goal is to minimize the cross-entropy loss between the prediction $\hat{y}_i$ and its true value $y_i$ on the training set $D^{\mathrm{tr}}$:

$$\min_{\{W,\theta\}} \mathcal{L}(x^{\mathrm{tr}}, y^{\mathrm{tr}}; W, \theta) = -\frac{1}{N_{\mathrm{tr}}} \sum_{i=1}^{N_{\mathrm{tr}}} \sum_{j=0}^{k-1} y_i^{\mathrm{tr}}(j) \cdot \log(\hat{y}_i^{\mathrm{tr}}(j)) \tag{3}$$

where $k$ denotes the number of lithology categories, $y_i \in \{0,1\}^k$ is the ground truth corresponding to $x_i$. $\hat{y}_i \in \{0,1\}^k$ is the prediction probability that $x_i$ belongs to category $j$. $y_i$ and $\hat{y}_i$ are both one-hot vectors.

For the output logits $X_i^{(l)}$, using the softmax function, the classifier $\hat{y}_i(j)$ is defined as:

$$\begin{cases} \hat{y}_i(j) = P(y = j | X_i^{(l)}, \theta_j) = \dfrac{e^{\langle X_i^{(l)}, \theta_j \rangle}}{\sum_{s=0}^{k-1} e^{\langle X_i^{(l)}, \theta_s \rangle}} \in \mathbb{R} \\ \theta = [\theta_0, \theta_1, \ldots, \theta_{k-1}] \end{cases} \tag{4}$$

where $\langle , \rangle$ denotes the inner product, and $\theta_j$ is the weight column vector learned by the classifier ($j \in \{0, 1, \ldots, k-1\}$).

$\mathcal{J}$ is the predicted label of the input image $x_i$. It equals the index of the category with the highest probability, defined by the following equation:

$$\mathcal{J} = \arg\max_j \{\hat{y}_i(j)\} \tag{5}$$

## 4. Methodology

In this section, we will develop a deep progressive learning model named RockNet for lithology recognition, based on local feature saliency and feature fusion. We will introduce the core components of the Multi-channel Feature Fusion (MFF) block and Local Feature Salience (LFS) block, respectively.

### 4.1. Overall architecture

RockNet consists of three main stages, namely category-aware augmentation, progressive learning, and multi-scale testing and prediction fusion, as shown in Fig. 4. RockNet exploits the consistent and complementary features from multi-view data while avoiding learning redundant representations. In addition, RockNet applies an efficient category-aware augmentation method to improve the contribution of minority categories. Moreover, RockNet includes MFF and LFS blocks. The MFF block captures and fuses hierarchical features from each view of rock images, while the LFS block extracts subtle information across different views.

### 4.2. Category-aware augmentation module

The rock image dataset has a relatively small amount of training data. Additionally, the dataset suffers from a serious category imbalance problem. To achieve a fair classifier for both majority and minority categories, we design a category-aware augmentation module in the RockNet model. It consists of a category-balanced mini-batch generator and global–local feature augmentation. Fig. 5 illustrates an example of category-aware augmentation.

**Algorithm 1** Category-aware augmentation algorithm

**Input:**
$k$: the number of categories in the dataset;
$A$: the number of mini-batches;
$s_i$: the image list of each category in the dataset, where $i = 0, 1, \ldots, k-1$.
**Output:**
$mini\_batch_b$, where $b = 1, 2, ..., A; list\_size$.
1: $n\_max \leftarrow$ the maximum of $s_i$;
2: **for** $i$ from 0 to $k-1$ **do**
3: Shuffle $s_i$; $l_i \leftarrow$ the size of $s_i$;
4: Sample $n_{max} - l_i$ images from $s_i$ randomly and append them to $s_i$ by Eq. (6);
5: $start \leftarrow 0$;
6: **for** $b$ from 1 to $A$ **do**
7: $stop \leftarrow start + \lfloor \frac{n\_max}{A} \rfloor$;
8: $c_{b0}$.append($s_i$[$start$: $stop$]);
9: $start \leftarrow stop$;
10: **end for**
11: **end for**
12: $N \leftarrow n\_max \times k$; $list\_size \leftarrow \lfloor \frac{N}{A} \rfloor$;
13: **for** $b$ from 1 to $A$ **do**
14: Performs auto augmentation for images in $c_{b0}$;
15: **for** $j$ from 0 to $list\_size - 1$ **do**
16: $c_{b1}[j], \ldots, c_{b4}[j] \leftarrow \mathrm{crop}(c_{b0}[j])$ by Eq. (8);
17: Perform auto-augmentation on the four images, by Eq. (9);
18: Resize them to $n \times n$ and then append them to lists $c_{b1}[j], \ldots, c_{b4}[j]$;
19: **end for**
20: Perform auto-augmentation, resize to $n \times n$ for images in $c_{b0}$;
21: $mini\_batch_b \leftarrow c_{b0}, c_{b1}, \ldots, c_{b4}$ by Eq. (10);
22: **end for**
23: $batch\_size \leftarrow 5 \times list\_size$;
24: **return** $mini\_batch_b, list\_size$.

#### 4.2.1. Category-balanced mini-batch generator

We design category-balanced oversampling to alleviate class imbalance. Instead of designing complex algorithms to synthesize new minority samples artificially, we directly copy minority samples to achieve category balance. The details of the category-balanced mini-batch generator are described in lines 1–11 of Algorithm 1.

We first obtain the maximum number of samples in $k$ categories and save the number as $n\_max$. We further shuffle each category and then perform random selection to ensure all $k$ categories attain the desired size $n_{max}$, defined by:

$$\begin{cases} s_i = \mathrm{shuffle}(s_i) \\ s_i = \mathrm{append}(\mathrm{random\_select}(s_i[j])) \end{cases} \tag{6}$$

where $s_i$ represents the image list for the $i$th category, and $j$ denotes the sample index within $s_i$. The operators shuffle(·), random_select(·), and append(·) correspond to shuffling, random selection, and appending, respectively. In Algorithm 1, specifically in lines 2–4, we first ascertain the size of $s_i$, which is represented by $l_i$. Subsequently, to ensure $s_i$ attains the desired size $n_{max}$, we oversample $n_{max} - l_i$ additional samples and append them to $s_i$. Following this oversampling, each category is guaranteed to contain $n_{max}$ samples, achieving balance across all $k$ categories. Consequently, the overall dataset size $N$ expands to $n_{max} \times k$.

In RockNet, a mini-batch is the fundamental data component in the model training procedure. To further alleviate the category imbalance

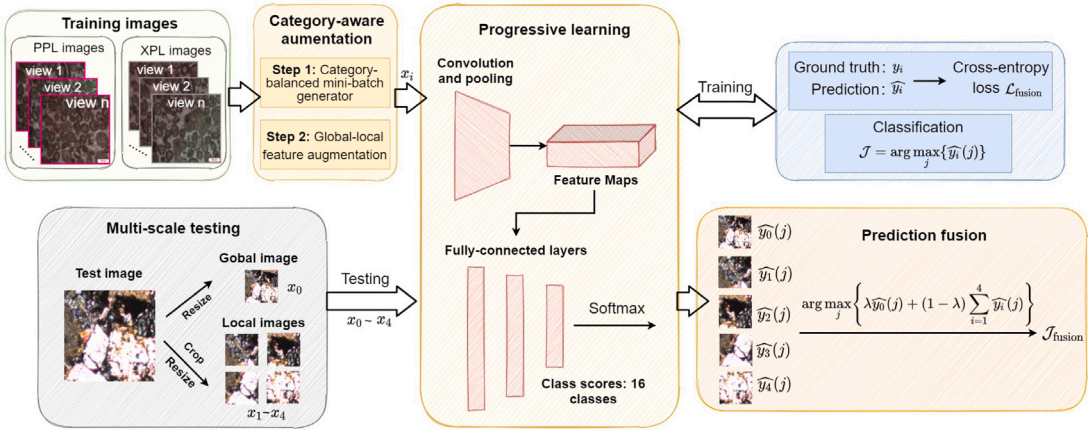



**Fig. 4.** RockNet's three-stage workflow: category-aware augmentation, progressive learning, and multi-scale prediction fusion.

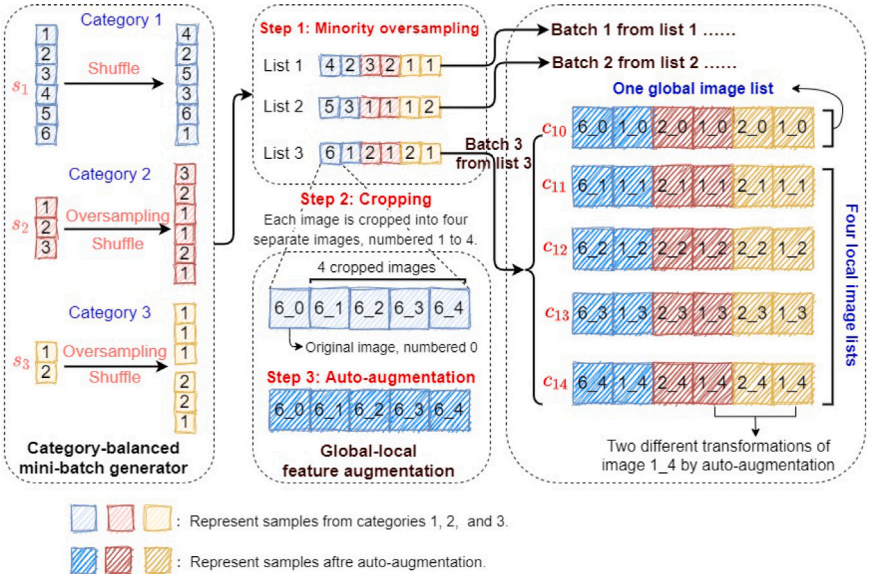



**Fig. 5.** Category-aware augmentation example: generating three thirty-sample mini-batches from categories with 6, 3, and 2 samples each. Each batch consists of one global image list and four local image lists. The global image list contains the original, uncropped images, while each of the four local image lists contains images that are cropped using a consistent method specific to that list.

problem, we design a category-balanced mini-batch generator. Given $A$ mini-batches, the key idea is to assign an equal number of samples from each category to construct a global image list, denoted as $c_{b0}$. This process is mathematically formulated as:

$$c_{b0} = \text{append}\left(s_i\left[j : j + \left\lfloor\frac{n\_max}{A}\right\rfloor\right]\right) \tag{7}$$

where $j$ ranges from 0 to $\left\lfloor\frac{n\_max}{A}\right\rfloor$, incremented by $\left\lfloor\frac{n\_max}{A}\right\rfloor$ each time, and $i$ ranges from 0 to $k-1$, representing the category index. The mini-batch index $b$ ranges from 1 to $A$. As described in lines 6–10 of Algorithm 1, we assign $\lfloor\frac{n\_max}{A}\rfloor$ samples from $s_i$ to $c_{b0}$. Consequently, $c_{b0}$ consists of an equal number of images, $\lfloor\frac{n\_max}{A}\rfloor$, selected from each of the $k$ categories. The length of $c_{b0}$ is $\lfloor\frac{n\_max\times k}{A}\rfloor$.

#### 4.2.2. Global–local feature augmentation

Rock micrographs are high-resolution images with a maximum resolution of $4908\times3264$ pixels and a minimum resolution of $800\times600$ pixels. To thoroughly capture local features, we develop a global–local feature augmentation method based on image cropping, augmentation, and resizing operations, as depicted in Fig. 6. We first split a rock image $c_{b0}[j]$ into four images, using a sliding window of size $\frac{3}{4}(h\times w)$ and a step of $\frac{1}{4}w$, where $w$ and $h$ represent the width and height of the image in pixels, respectively. Accordingly, the cropping operation is defined as:

$$c_{b1}[j], c_{b2}[j], c_{b3}[j], c_{b4}[j] = \text{crop}(c_{b0}[j]) \tag{8}$$

where $j$ is the image index within $c_{b0}$, $j = 0, 1, \ldots, \lfloor\frac{n\_max\times k}{A}\rfloor - 1$, $b$ is the mini-batch index, and $b = 1, 2, \ldots A$. The four local images obtained from Eq. (8) are saved to the corresponding local image list $c_{b1}$, $c_{b2}$, $c_{b3}$, and $c_{b4}$.

To reduce the negative impact of oversampling, we apply a simple auto-augmentation to perform data augmentation on images randomly. This augmentation method is encapsulated in the following equation:

$$c_{bi}[j] = \text{auto-augmentation}(c_{bi}[j]) \tag{9}$$

where $i$ represents the index of the image lists, $i = 0, 1, \ldots, 4$, $j$ ranges from 0 to $\lfloor\frac{n\_max\times k}{A}\rfloor - 1$, and $b$ ranges from 1 to $A$, denoting the mini-batch index.

Auto-augmentation, an online and random image transformation strategy, uses a search algorithm to find the best augmentation strategy to achieve the highest validation accuracy on the dataset. It is available in the *torchvision.transforms* module of the PyTorch framework and includes fourteen augmentation strategies [30].

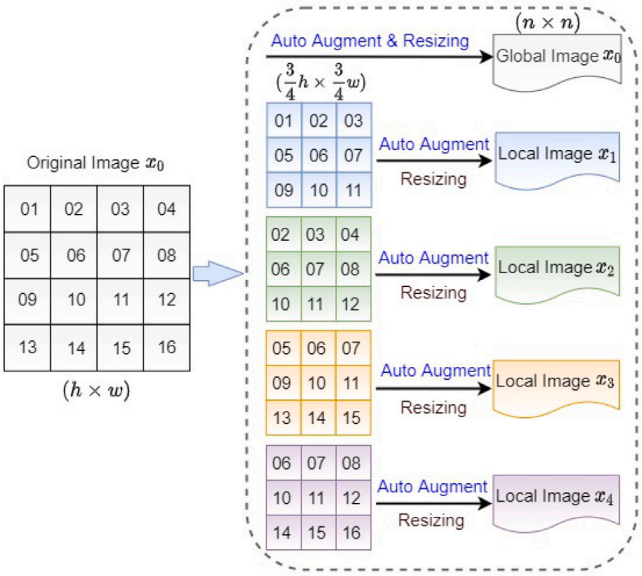



**Fig. 6.** Global–local feature construction method illustration.

Due to insufficient graphics processing unit (GPU) memory, it is infeasible to directly input original images into our RockNet model for training. Therefore, all images in $c_{bi}$ are resized to $n \times n$ pixels to form a mini-batch, and then input to RockNet for effective training, as defined in Eq. (10).

$$mini - batch_b = \text{resize}(c_{bi}) \tag{10}$$

where $i = 0, 1, \ldots, 4$, and $b = 1, 2, \ldots A$.

The global–local feature augmentation, as detailed in lines 13–22 of Algorithm 1, offers several key advantages. By extracting and transforming cropped images from the same original image, we introduce a variety of transformations that, although derived from a single source, contribute to a rich and diverse training dataset. This diversity is crucial for effectively generalizing the model and improving its robustness.

Furthermore, the augmentation process results in a category-balanced mini-batch $mini\_batch_b$, achieved by skillfully combining five distinct image lists $c_{bi}$. This approach not only ensures a balanced representation of categories but also increases the batch size to five times the number of images in the list $c_{b0}$, as indicated in line 23 of Algorithm 1. This enlargement of the batch size is instrumental in leveraging more information from each training example.

Importantly, the global–local feature enhancement leads to a non-uniform frequency distribution of the original 16 image patches across the four cropped images, as shown in Fig. 6. Some patches, such as 01, 04, 13, and 16, appear only once, while others like 06, 07, 10, and 11 are presented four times. This disparity in frequency is intentional, emphasizing the significance of local features. Patches that appear more frequently are subjected to a higher number of image transformations, which amplifies the local information and, consequently, enhances their impact on the training loss.

In summary, the global–local feature augmentation technique enriches the training process by diversifying the data, emphasizing the significance of local features, and optimizing the batch size for more efficient model training.

### 4.3. Model construction

Like ordinary deep neural networks, RockNet consists of an input layer, feature extractor, and classifier. Fig. 7(a) shows the structure of RockNet. The input is $224 \times 224 \times 3$ thin section images. The highlight of the feature extractor is the block design of MFF and LFS. The initial hidden layer is a convolutional block with 32 filters. The filter size, stride, and padding are $3 \times 3$, 1, and 1, respectively, and the output is $224 \times 224 \times 32$. The subsequent hidden layers of the network are composed of alternating MFF and LFS blocks, with each MFF block being immediately followed by an LFS block, creating a structured pattern that repeats throughout the feature extraction process. In the classifier, following the fifth LFS block, an adaptive pooling layer is applied to the output, reducing the spatial dimensions while retaining essential features. This is followed by two fully connected layers with 1024 neurons each, designed to capture complex patterns within the feature space. The final layer has 16 neurons, corresponding to the number of lithology categories recognized by the RockNet model. To improve the training speed and overall performance, RockNet adds BN and applies SiLU as the activation function. Given an input image $x_i$, the feature representation $X_i^{(1)}$ of the first hidden layer is defined as:

$$\begin{aligned} X_i^{(1)} &= \text{SiLU}(\text{BN}(f^{(1)}(x_i; W^{(1)}))) \\ &= \text{SiLU}(\text{BN}(\text{Conv2D}(3, 32, 3, 1, 1))) \end{aligned} \tag{11}$$

where 3 is the channel amount of $x_i$, 32 is the channel amount generated by the convolution, 3 is the filter size of the convolution, and 1 and 1 are the stride and padding of the convolution, respectively.

#### 4.3.1. Multi-channel feature fusion block

We construct the Multi-channel Feature Fusion (MFF) block to extract and fuse hierarchical features from each view of rock images, as shown in Fig. 7(b). Inspired by "Inception-v4", we adopt a flexible multi-way and multi-scale feature fusion strategy in RockNet. It performs four-path convolutions in parallel and forms the final feature map by taking the element-wise sum of these outputs. Assume that the input feature map is $h \times w \times dim$, where $w$ and $h$ are the width and height in pixels, and $dim$ is the channel amount. After performing MFF, the output will be $\frac{1}{2}h \times \frac{1}{2}w \times 2dim$. The MFF block halves the spatial dimension and doubles the depth of the feature map. Given a feature map $X^{(j-1)}$, the MFF block is formulated as:

$$\begin{aligned} &\text{MFF}^{(j)}(X_i^{(j-1)}; W^{(k)}) = \\ &f_A^{(j)}(X_i^{(j-1)}; \Theta_1^{(k)}) + f_C^{(j)}(f_B^{(j)}(X_i^{(j-1)}; \Theta_2^{(k)}); \Theta_3^{(k)}) + \\ &f_D^{(j)}(X_i^{(j-1)}; \Theta_4^{(k)}) + f_E^{(j)}(f_D^{(j)}(X_i^{(j-1)}; \Theta_4^{(k)}); \Theta_5^{(k)}) \end{aligned} \tag{12}$$

where $f^{(j)}(\cdot)$ denotes the Conv2D operation, as defined in Eq. (2), using weights $W^{(k)}$, and superscripts $j$ and $k$ represent the indices of hidden layers and convolutional operations, respectively. The subscripts $A, \ldots, E$ correspond to the indices of the convolutions within the MFF block, listed from left to right. Their corresponding weights are $\Theta_1^{(k)}, \ldots, \Theta_5^{(k)}$. It should be noted that the weight matrix for the $j$th layer is $W^{(k)} = \{\Theta_1^{(k)}, \ldots, \Theta_5^{(k)}\}$.

The MFF block increases the receptive field while decreasing the model parameters by combining convolution filters of varying sizes into a large filter. Specifically, from left to right in the MFF block, we sequentially stack two non-linear convolution layers, namely convolutions B and C in the second path, and convolutions D and E in the fourth path. Compared to a single convolution operation, the four parallel paths have receptive fields of 3, 3, 5, and 9, respectively, thereby increasing the network's discriminative power. Notably, the stacked $3 \times 3$ and $5 \times 5$ convolutional layers in the fourth path have an effective receptive field of $9 \times 9$. However, these layers are parameterized more efficiently than a single $9 \times 9$ convolutional layer. The $5 \times 5$ convolutional layer has $5 \times 5 \times dim \times 2dim = 50dim^2$ weights, and the $3 \times 3$ convolutional layer has $3 \times 3 \times 2dim \times 2dim = 36dim^2$ weights, totaling $86dim^2$. In contrast, a single $9 \times 9$ convolutional layer would require $162dim^2$ weights, which is 88% more. Additionally, a $1 \times 1$ convolutional layer is added to the second path, effectively increasing the network's non-linearity.

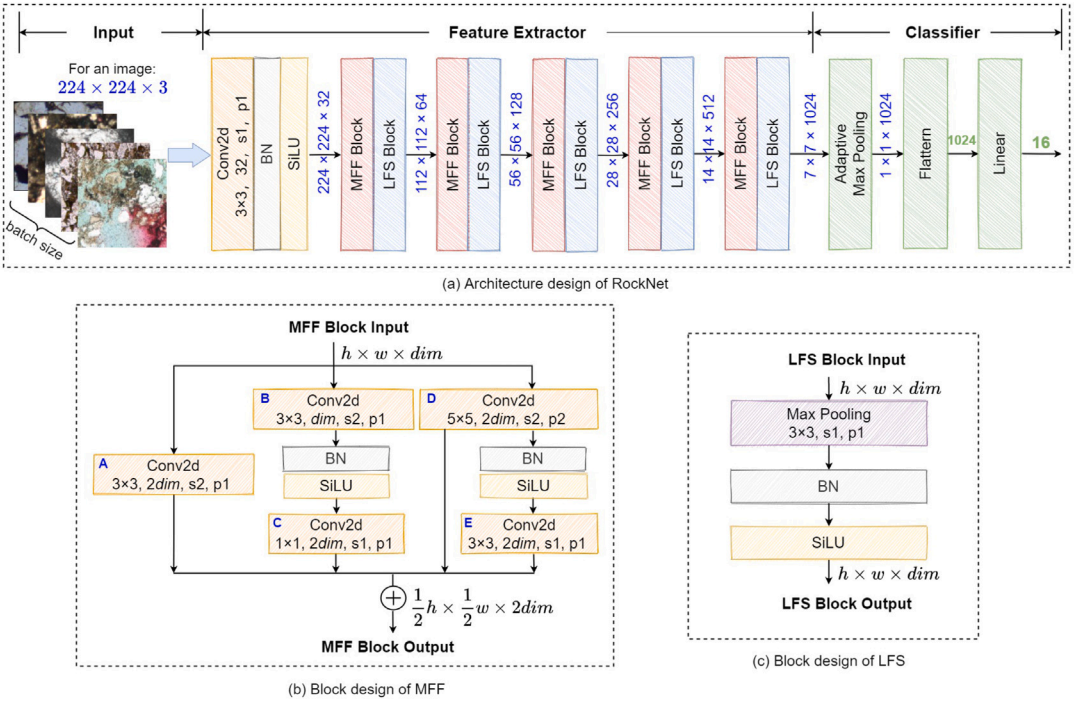



**Fig. 7.** RockNet model architecture: core components of Multi-channel Feature Fusion (MFF), Local Feature Salience (LFS), and classifier.

#### 4.3.2. Local feature salience block

We design the Local Feature Salience (LFS) block to extract subtle information across different views, maintaining the output dimension identical to the input. Unlike traditional pooling layers, LFS preserves spatial resolution, which is crucial since thin sections contain more details and valuable information that would be lost with downsampling through pooling operations. As shown in Fig. 7(c), the LFS block performs two-dimensional max pooling (MaxPool2D) on a 3 × 3 pixel window with a stride of 1 and padding of 1 to expand local feature saliency without reducing the spatial dimensions. In addition, we apply both BN and SiLU sequentially after max pooling, instead of applying them to each path within the MFF block. This approach further reduces computational costs while maintaining training effectiveness. The calculation process of the LFS block is defined as:

$$\begin{aligned}&\mathrm{LFS}^{(j)}(X_i^{(j-1)}, r, s, p) = \\ &\quad \mathrm{SiLU}(\mathrm{BN}(\mathrm{MaxPool2D}(X_i^{(j-1)}, 3, 1, 1)))\end{aligned} \tag{13}$$

where $j$ denotes the hidden layer index for $j \in \{1, 2, \ldots, l\}$. $X_i^{(j-1)}$ represents the output features from the layer immediately preceding the $j$th layer in the network. In the MaxPool2D operation, the parameters are a pool size $r$ of 3, a stride $s$ of 1, and a padding $p$ of 1, respectively.

### 4.4. Progressive training strategy

Motivated by $k$-fold cross-validation, we propose a $k$-fold progressive training strategy to determine the optimal model parameters, where $k$ is a hyperparameter. Initially, the dataset is partitioned into three segments: 60% for training, 20% for validation, and 20% for testing, respectively. The test set is used for final evaluation. The remaining data is shuffled into $k$ different combinations of training and validation sets. The $k$-fold progressive training workflow is shown in Fig. 8. In each quarter of the training epochs, the process is carried out in two distinct yet interconnected steps for each of the four folds: First, a model is trained using three of the folds as training data. Subsequently, the model that emerges from this training phase is validated on the remaining part of the data.

We adopt the general progressive concept to fine-tune RockNet. The training cycle is divided into four stages, as depicted in lines 2–3 of Algorithm 2. In each cycle, different training and validation data will be applied. The parameters will be continuously adjusted according to the previous learning process in the next cycle. The whole model training is a progressive learning process. Furthermore, the $k$-fold progressive training can be regarded as a data-diversifying strategy, forcing the model to learn from more data. It can gradually enhance the capacity for more general and richer feature representation. Therefore, the generalization capability of RockNet can be enhanced.

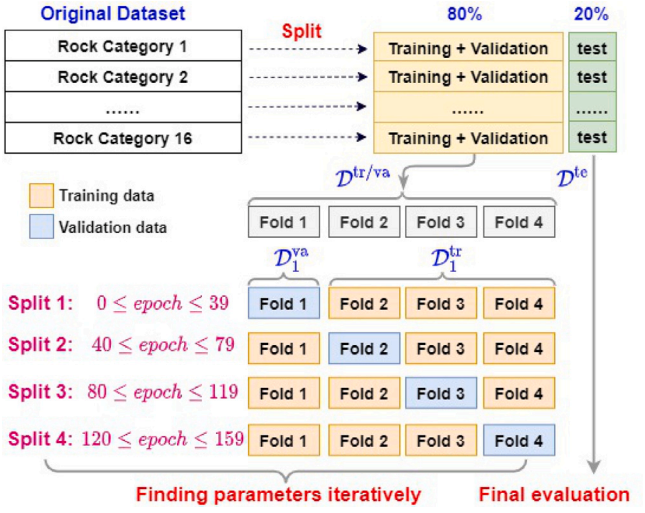



**Fig. 8.** Four-fold progressive training workflow.

To emphasize the significance of both global and local features in our model, we introduce a novel loss function, as defined by Eq. (3).

$$\mathcal{L}_{\text{fusion}} = \lambda \times \mathcal{L}_{c_{b0}} + (1 - \lambda) \times \sum_{i=1}^{4} \mathcal{L}_{c_{bi}} \tag{14}$$

where $\lambda$ represents the loss contribution factor, and $b$ indexes the mini-batches from 1 to $A$. The term $\mathcal{L}_{c_{b0}}$ captures the loss associated with the

global images in $c_{b0}$, ensuring that the broader context is considered during training. Conversely, the summation term $\sum_{i=1}^{4}\mathcal{L}_{c_{bi}}$ represents the aggregated loss from local features, with each $c_{bi}$ corresponding to the $i$th image list of local images. The index $i$ ranges from 1 to 4, corresponding to different local regions or features extracted from the global image. The value of $\lambda$ is crucial as it modulates the balance between the global and local components of the loss, thus influencing the learning process and the model's ability to generalize from the training data.

The detailed steps for the progressive training process are outlined in Algorithm 2. Within this algorithm, the parameters $W$ and $\theta$ of RockNet are optimized through lines 10–30.

**Algorithm 2** Progressive model training

**Input:**
$D$: the dataset; $e$: the number of epochs;
$\lambda$: the loss contribution factor; $\eta$: the learning rate.
**Output:**
The weights $W$ and $\theta$.
1: Initialize $W$ and $\theta$ randomly;
2: $\{D^{\text{tr/va}}, D^{\text{te}}\} \leftarrow \text{Split}(D)$, with the ratio of 8:2;
3: $\{Fold_1, \ldots, Fold_4\} \leftarrow \text{Split}(D^{\text{tr/va}})$. Combine different three folds to construct four training sets $D_1^{\text{tr}}, \ldots, D_4^{\text{tr}}$. The corresponding left fold constructs the validation sets $D_1^{\text{va}}, \ldots, D_4^{\text{va}}$, as described in Fig. 8;
4: $e \leftarrow 160$; $\eta \leftarrow 0.0001$; $s \leftarrow 1$;
5: **for** *epoch* from 1 to $e$ **do**
6: **if** $epoch$ % 40 == 0 **then**
7: By Algorithm 1, perform category-aware augmentation over ($D_s^{\text{tr}}$) to obtain $A$ mini-batches. Each mini-batch comprises five image lists, denoted as $c_{bi}$, with the size of $list\_size$;
8: $s \leftarrow s + 1$;
9: **end if**
10: **for** $b$ from 1 to $A$ **do**
11: **for** $i$ from 0 to 4 **do**
12: $\mathcal{L}_{c_{bi}} \leftarrow 0$; $j \leftarrow 1$; $k \leftarrow 1$;
13: **for** $l$ from 1 to $list\_size$ **do**
14: $X_l^{(j)} \leftarrow \text{SiLU}(\text{BN}(f^{(j)}(x_l; W^{(k)})))$ by Eq. (11); $k \leftarrow k + 1$;
15: **for all** five MFF blocks **do**
16: $X_l^{(j+1)} \leftarrow \text{MFF}^{(j+1)}(X_l^{(j)}; W^{(k)})$ by Eq. (12); $k \leftarrow k + 1$;
17: $X_l^{(j+2)} \leftarrow \text{LFS}^{(j+2)}(X_l^{(j+1)}, 3, 1, 1)$ by Eq. (13); $j \leftarrow j + 2$;
18: **end for**
19: $X_l^{(12)} \leftarrow \text{AdaptiveMaxPool2D}(X_l^{(11)}, (1, 1))$;
20: $X_l^{(13)} \leftarrow \text{Flatten}(X_l^{(12)})$;
21: $X_l^{(14)} \leftarrow \text{Linear}(X_l^{(13)})$;
22: $\hat{y}_l \leftarrow \text{softmax}(X_l^{(14)}; \theta)$ by Eq. (4);
23: Compute $\mathcal{L}_{x_l}$ for image $x_l$ by Eq. (3);
24: $\mathcal{L}_{c_{bi}} += \mathcal{L}_{x_l}$
25: **end for**
26: **end for**
27: Compute $\mathcal{L}_{\text{fusion}}$ by Eq. (14);
28: Update $W$ and $\theta$ by Eq. (18);
29: **end for**
30: **end for**
31: **return** $W$ and $\theta$.

### 4.5. Multi-scale prediction fusion module

In the inference process, we further design a multi-scale prediction fusion strategy for the well-trained RockNet model. Given an input validation or testing image $x_0$, the inference consists of three steps. Firstly, through global–local feature augmentation, $x_0$ is split into four local images $(x_1, \ldots, x_4)$. We then perform auto augmentation and resizing on $x_i$, where $i = 0, \ldots, 4$. Next, $x_i$ is fed to RockNet to obtain its final feature representation $X_i^{(l)}$, and then five predictions $\hat{y}_i$ are realized based on Eq. (4). Finally, weights $\lambda$ and $1 - \lambda$ are assigned to $\hat{y}_0$ and the remaining four predictions.

The final prediction $\mathcal{J}_{\text{fusion}}$ is obtained by the weighted sum of the five predictions, with each prediction contributing uniquely to the final result, reflecting their importance in the model. The calculation process of the multi-scale inference is defined as:

$$\mathcal{J}_{\text{fusion}} = \arg\max_j \left\{ \lambda \hat{y}_0(j) + (1-\lambda) \sum_{i=1}^{4} \hat{y}_i(j) \right\} \tag{15}$$

where $\lambda$ is the loss contribution factor that highlights the importance of the global prediction. The term $\hat{y}_0(j)$ represents the probability that the global image $x_0$ is predicted to belong to category $j$. Conversely, the summation term $\sum_{i=1}^{4} \hat{y}_i(j)$ represents the aggregated probability from the four predictions of the local images. The detailed steps of the multi-scale prediction fusion module are described in Algorithm 3.

**Algorithm 3** Multi-scale prediction fusion module

**Input:**
$x_0$: a validation or a testing image;
$\lambda$: the loss contribution factor;
$W, \theta$: the weights achieved from Algorithm 2.
**Output:**
The rock category index $\mathcal{J}_{\text{fusion}}$ of $x_0$.
1: Cut $x_0$ into four images $x_1, \ldots, x_4$ by using the global-local feature augmentation;
2: **for** $i$ from 0 to 4 **do**
3: Perform auto-augmentation and resizing for $x_i$;
4: Calculate the feature representation $X_i^{(14)}$ for $x_i$ by using lines 14–21 of Algorithm 2;
5: Calculate the prediction $\hat{y}_i$ by using line 22 of Algorithm 2;
6: **end for**
7: Calculate the final prediction $\mathcal{J}_{\text{fusion}}$ for $x_0$ by using Eq. (15);
8: **return** The predicted index $\mathcal{J}_{\text{fusion}}$

### 4.6. Model parameter configuration

In this section, we describe the configuration of model parameters for RockNet. As illustrated in Fig. 7, RockNet is composed of various layers, each with its own set of parameters. Specifically, the model parameters of RockNet are categorized into three main components:

- The weight matrix $W^{(1)}$ of the first hidden layer, which serves as the input to the subsequent layers.
- The set of weight matrices for the MFF block, denoted as $W^{(j)}$ for $j = 2, \ldots, 6$. Each $W^{(j)}$ comprises five weight matrices $\Theta_1^{(j)}, \ldots, \Theta_5^{(j)}$ that correspond to different paths within the MFF block.
- The weight matrix $\theta$ of the output layer, which maps the final features to the prediction scores for each category.

The parameters are defined as follows:

$$\begin{cases} W &= [W^{(1)}, W^{(2)}, \ldots, W^{(6)}] \\ \theta &= [\theta_0, \theta_1, \theta_2, \ldots, \theta_{15}] \end{cases} \tag{16}$$

where $W^{(j)} = \{\Theta_1^{(j)}, \ldots, \Theta_5^{(j)}\}$ for $j = 2, \ldots, 6$. $\Theta_i$ represents the weight matrices for the $i$th convolutional operation of an MFF block. $\theta_k$ is the weight column vector learned by the classifier for $k \in \{0, 1, \ldots, 15\}$.

The dimensions of these parameters can be represented as:

$$\begin{cases} W^{(1)} \in \mathbb{R}^{32\times3\times3\times3} \\ \Theta_1 \in \mathbb{R}^{2dim\times3\times3\times dim} \\ \Theta_2 \in \mathbb{R}^{dim\times3\times3\times dim} \\ \Theta_3 \in \mathbb{R}^{2dim\times1\times1\times dim} \\ \Theta_4 \in \mathbb{R}^{2dim\times5\times5\times dim} \\ \Theta_5 \in \mathbb{R}^{2dim\times3\times3\times2dim} \\ \theta \in \mathbb{R}^{1024\times16} \end{cases} \tag{17}$$

where $\Theta_i$ for $i = 1, \ldots, 5$ has dimensions $\mathbb{R}^{F_i\times H_i\times W_i\times C_i}$ with $F_i$ being the number of filters, $H_i$ and $W_i$ being the height and width of the filters, and $C_i$ being the number of input channels. The term $dim$ denotes the base number of channels, and $2dim$ represents twice the number of channels. The term $W^{(1)}$ represents the weights of the first hidden layer, with dimensions $\mathbb{R}^{32\times3\times3\times3}$ indicating 32 filters of size $3 \times 3$ on 3 input channels. The term $\theta$ represents the weights of the output layer, with dimension $\mathbb{R}^{1024\times16}$, where 1024 corresponds to the size of the final logits of the last hidden layer, and 16 is the number of output categories.

The optimal weights $W$ and $\theta$ of RockNet can be found by minimizing the loss function $\mathcal{L}_{\text{fusion}}$ using gradient descent. The update rule for the parameters is given by:

$$\begin{cases} W^{(j)} = W^{(j-1)} - \eta \cdot \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial W}\Big|_{W=W^{(j-1)}} \\ \theta^{(k)} = \theta^{(k-1)} - \eta \cdot \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \theta}\Big|_{\theta=\theta^{(k-1)}} \end{cases} \tag{18}$$

where $\eta$ is the learning rate, $W^{(j)}$ represents the $j$th weights for $j = 1, \ldots, 6$, and $\theta$ represents the output layer weights for $k = 0, \ldots, 15$. The error propagation gradient of the $i$th hidden layer is defined as:

$$\begin{cases} \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial W^{(j)}} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f^{(i)}} \times \dfrac{\partial f^{(i)}}{\partial W^{(j)}} \\ \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \theta} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \hat{y}} \times \dfrac{\partial \hat{y}}{\partial \theta} \end{cases} \tag{19}$$

where $j = 1, \ldots, 6$, $f^{(i)}$ denotes the convolutional operation of the $i$th hidden layer, and $\hat{y}$ represents the predicted output of the network.

Based on Eq. (12) and the chain rule, the error propagation gradient of the parameters $\Theta$ of the MFF block is defined as follows:

$$\begin{cases} \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \Theta_1^{(r)}} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f_A} \times \dfrac{\partial f_A}{\partial \Theta_1^{(r)}} \\ \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \Theta_2^{(r)}} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f_C} \times \dfrac{\partial f_C}{\partial f_B} \times \dfrac{\partial f_B}{\partial \Theta_2^{(r)}} \\ \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \Theta_3^{(r)}} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f_C} \times \dfrac{\partial f_C}{\partial \Theta_3^{(r)}} \\ \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \Theta_4^{(r)}} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f_D} \times \dfrac{\partial f_D}{\partial \Theta_4^{(r)}} + \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f_E} \times \dfrac{\partial f_E}{\partial f_D} \times \dfrac{\partial f_D}{\partial \Theta_4^{(r)}} \\ \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial \Theta_5^{(r)}} = \dfrac{\partial \mathcal{L}_{\text{fusion}}(W,\theta)}{\partial f_E} \times \dfrac{\partial f_E}{\partial \Theta_5^{(r)}} \end{cases} \tag{20}$$

where $r$ denotes the layer index for $r = 2, \ldots, 6$, $f_A, f_B, \ldots, f_E$ represent the Conv2D operations specific to the four paths of the MFF block.

## 5. Experiments

We perform rigorous experiments to evaluate the performance of RockNet. We first compare RockNet with existing state-of-the-art methods. Then, we verify the effectiveness of the proposed category-aware enhancement and progressive training strategies. Finally, we discuss the impact of the loss contribution factors and determine their optimal values.

**Table 1**
Number of original datasets.

| Rock type | XPL image | PPL image | Training set | Validation set | Test set |
|---|---|---|---|---|---|
| F | 56 | 56 | 68 | 22 | 22 |
| FQ | 238 | 238 | 286 | 95 | 95 |
| FL | 23 | 23 | 28 | 9 | 9 |
| Q | 337 | 334 | 403 | 134 | 134 |
| QL | 218 | 218 | 262 | 87 | 87 |
| L | 15 | 15 | 18 | 6 | 6 |
| LF | 44 | 30 | 45 | 15 | 14 |
| LQ | 74 | 74 | 89 | 30 | 29 |
| Oolitic | 97 | 97 | 117 | 39 | 38 |
| Grainstone | 153 | 148 | 181 | 60 | 60 |
| Wackestone | 130 | 131 | 157 | 52 | 52 |
| Mudstone | 180 | 180 | 216 | 72 | 72 |
| Packstone | 112 | 112 | 135 | 45 | 44 |
| Floatstone | 123 | 123 | 148 | 49 | 49 |
| Arenaceous | 81 | 81 | 98 | 32 | 32 |
| Microcrystal | 230 | 230 | 276 | 92 | 92 |
| Total | **2111** | **2090** | **2527** | **839** | **835** |

### 5.1. Dataset and experimental settings

#### 5.1.1. Dataset

We collect the dataset from an open database of rock micrographs [31]. The dataset contains 4201 images of two major sedimentary rocks, specifically limestone and sandstone, as detailed in Table 1.

The dataset includes plane-polarized and cross-polarized light images of rock-thin sections. According to the simplified Garzanti classification, sandstone is further subdivided into 8 secondary rock types, namely feldspathic (F), feldspatho-quartzose (FQ), feldspatho-lithic (FL), quartzose (Q), quartzo-lithic (QL), lithic (L), litho-feldspathic (LF) and litho-quartzose (LQ), as shown in Figs. 9(a) to 9(h). Limestone is further divided into 8 secondary rock types according to the modified Dunham classification, namely oolitic, grainstone, wackestone, mudstone, packstone, floatstone, arenaceous, and microcrystal, as shown in Figs. 9(i) to 9(p).

The dataset is randomly partitioned into segments: 60% for training, 20% for validation, and 20% for testing. As shown in Table 1, the number of images in the training, validation, and test sets is 2527, 839, and 835, respectively.

#### 5.1.2. Experimental settings

We perform all comparison experiments on a server equipped with an Intel(R) Xeon(R) Gold 6330 processor and three NVIDIA A100 GPUs. The operating system is Ubuntu 20.04. The programming framework is Pytorch 1.12.0. The version of the compute unified device architecture (CUDA) is 11.3. To ensure a fair evaluation, the proposed category-aware augmentation and progressive learning strategies are applied to the state-of-the-art models. Moreover, all comparative methods employ the same training hyper-parameters as RockNet. The training epochs and batch size are both 160. The original learning rate and decay factor are 0.001 and 0.1, respectively. The optimizer is stochastic gradient descent (SGD). Except for ResNeSt-101, the input image size is $224 \times 224$ pixels.

### 5.2. Experimental results

We evaluate the recognition performance of our RockNet model on a comprehensive dataset encompassing sixteen distinct rock categories. The recognition performance of RockNet on the test set is summarized in Table 2.

Table 2 depicts that 75% of the rock categories can be well detected, with precision and recall exceeding 80%. In particular, RockNet can accurately classify QL with an F1 score and specificity of 100%. However, RockNet cannot identify FL, wackestone, and LF well, with

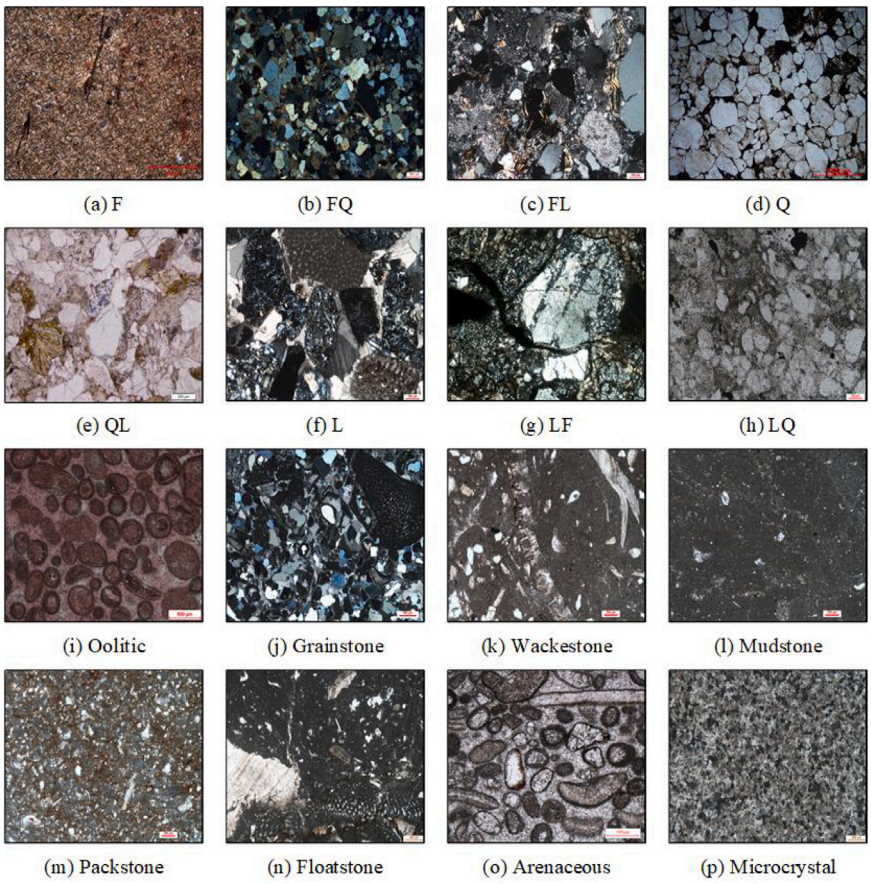



**Fig. 9.** Rock photomicrograph examples from the dataset: (a), (b), (c), (f), (g), and (j) are cross-polarized light (XPL) images, the rest are plane-polarized light (PPL) images.

**Table 2**
Recognition performance of RockNet on the test set.

| Rock type | Precision (%) | Recall (%) | F1 score (%) | Specificity (%) |
|---|---|---|---|---|
| F | 76.9 | 90.9 | 83.3 | 99.3 |
| FQ | 99.0 | 100.0 | 99.5 | 99.9 |
| FL | 100.0 | 44.4 | 61.5 | 100.0 |
| Q | 100.0 | 98.5 | 99.2 | 100.0 |
| QL | **100.0** | **100.0** | **100.0** | **100.0** |
| L | 100.0 | 83.3 | 90.9 | 100.0 |
| LF | 80.0 | 57.1 | 66.6 | 99.8 |
| LQ | 82.9 | 100.0 | 90.7 | 99.3 |
| Oolitic | 81.0 | 89.5 | 85.0 | 99.0 |
| Grainstone | 98.4 | 100.0 | 99.2 | 99.9 |
| Wackestone | 71.1 | 51.9 | 60.0 | 98.6 |
| Mudstone | 77.2 | 84.7 | 80.8 | 97.6 |
| Packstone | 83.7 | 93.2 | 88.2 | 99.0 |
| Floatstone | 82.7 | 87.8 | 85.2 | 98.9 |
| Arenaceous | 76.7 | 71.9 | 74.2 | 99.1 |
| Microcrystal | 94.4 | 91.3 | 92.8 | 99.3 |

lower recalls of 44.4%, 51.9%, and 57.1%, respectively. This is because sandstones are classified in the order of abundance of the main components feldspar (F), lithic (L), and quartz (Q). FL is rich in lithic fragments, but the feldspar content is higher than quartz, that is, L>F>Q. Wackestone is a mud-supported carbonate rock with more than 10% of the grains [32]. Sand-sized grains are usually composed of rock fragments, such as feldspar and quartz. Therefore, the key to accurately identifying FL, LF, and wackestone is to calculate the contents of F, L, and Q, which is more challenging than other rock categories.

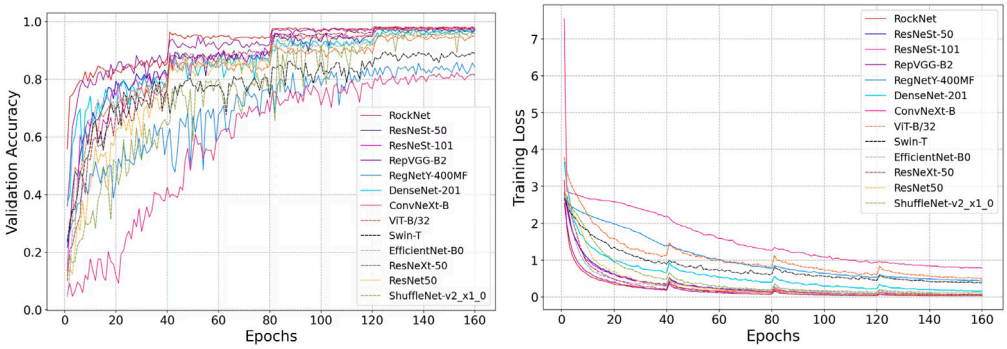
### 5.3. Performance evaluation

To provide a comprehensive assessment of recognition performance, we compare RockNet with ten prevalent CNN-based methods: EfficientNet [33], ResNeSt [34], ResNet [35], ResNeXt [36], DenseNet [37], RepVGG [38], RegNet [39], ShuffleNet [40], ConvNeXt [41], as well as two transformer-based models, ViT [42] and Swin-T networks [43]. In addition, model size and Giga floating point of operations (GFLOPs) are crucial metrics for evaluating deep learning models. The architectural effectiveness and recognition performance of RockNet relative to 12 prevalent models are detailed in Table 3.

In Table 3, columns 3 and 4 represent a comparison of model efficiency. RockNet demonstrates a modest computational demand, requiring only 7.5 GFLOPs and having a relatively moderate number of parameters at 40.2 million. As a medium-sized CNN model, RockNet possesses fewer parameters compared to models such as RepVGG-B2, ResNeSt-101, ConvNeXt-B, and ViT. Columns 5 to 9 show the recognition performance. RockNet excels in recognition performance across various metrics, including accuracy, precision, recall, F1 score, and specificity. Specifically, RockNet achieves the highest scores in accuracy and F1 score, reaching 90.1% and 87.4%, respectively. As discussed in Section 4, the task of rock lithology recognition is characterized by an imbalanced dataset. The F1 score, which emphasizes the balance between precision and recall, is thus a more comprehensive measure of performance than accuracy alone, particularly in such scenarios. Additionally, RockNet exhibits the highest specificity at 99.3%, indicating its exceptional capability in accurately identifying negative rock samples. This high specificity, coupled with its top performance in other metrics, underscores RockNet's effective balance between model accuracy and complexity.

**Table 3**
Comparative analysis of RockNet with 12 prevalent models: focus on architectural effectiveness and recognition performance.

| Model | Input image size | Model size (M) | GFLOPs | Accuracy (%) | Precision (%) | Recall (%) | F1 score (%) | Specificity (%) |
|---|---|---|---|---|---|---|---|---|
| RockNet | $224^2$ | 40.2 | 7.5 | **90.1** | **89.0** | **85.8** | **87.4** | **99.3** |
| EfficientNet-B0 | $224^2$ | 4.0 | 0.4 | 88.6 | 85.4 | 80.9 | 83.1 | 99.2 |
| ResNeSt-50 | $224^2$ | 25.5 | 5.4 | 87.2 | 85.9 | 80.0 | 82.8 | 99.2 |
| ResNeSt-101 | $256^2$ | 46.3 | 13.4 | 87.8 | 86.7 | 79.1 | 82.7 | 99.2 |
| ResNet50 | $224^2$ | 23.5 | 4.1 | 85.9 | 83.4 | 77.6 | 80.4 | 99.1 |
| ResNeXt-50 | $224^2$ | 23.0 | 3.8 | 88.0 | 83.6 | 77.3 | 80.3 | 99.2 |
| DenseNet-201 | $224^2$ | 18.1 | 4.4 | 86.1 | 82.1 | 78.2 | 80.1 | 99.1 |
| RepVGG-B2 | $224^2$ | 86.5 | 20.5 | 84.6 | 83.8 | 75.0 | 79.2 | 99.0 |
| RegNetY-400MF | $224^2$ | 3.9 | 0.4 | 83.0 | 79.5 | 75.5 | 77.4 | 98.9 |
| ShuffleNet-v2_X1_0 | $224^2$ | **1.3** | **0.2** | 85.4 | 81.2 | 77.6 | 79.4 | 99.0 |
| ConvNeXt-B | $224^2$ | 87.6 | 15.4 | 74.7 | 65.7 | 66.0 | 65.8 | 98.3 |
| ViT-B/32 | $224^2$ | 87.5 | 4.4 | 82.2 | 74.1 | 71.1 | 72.6 | 98.8 |
| Swin-T | $224^2$ | 27.5 | 4.5 | 82.3 | 76.2 | 71.6 | 73.8 | 98.8 |




(a) Validation accuracy (b) Training loss

**Fig. 10.** Validation accuracy and training loss: comparative analysis of RockNet with 12 prevalent models.

### 5.4. Accuracy and convergence analysis

We discuss the accuracy and convergence of the compared models. The experimental results of validation accuracy and training loss for these models are depicted in Fig. 10.

We can see from Fig. 10(a) that RockNet achieves the highest accuracy over other compared models. The best and average validation accuracy of RockNet are 98.28% and 93.04% respectively, which are 19.37% and 57.75% higher than those of ConvNeXt-B. RepVGG-B2 secures the second place, while ConvNeXt-B is the least performing. Notably, a significant improvement in accuracy is observed for most models at the 40th, 80th, 120th, and 160th epochs. These results demonstrate the effectiveness of the $k$-fold progressive learning strategy, which is characterized by periodic updates to the training and validation sets every 40 epochs, thereby gradually boosting the feature representation capability of the models. Furthermore, Fig. 10(b) shows that RockNet exhibits superior convergence performance, evidenced by a smooth and stable training loss curve.

### 5.5. Visualization of recognition results

To verify the performance of RockNet, we select five highly competitive models for comparison: EfficientNet-B0, ResNeSt-50, ResNet50, DenseNet-201, and RepVGG-B2, and visualize their recognition results. We also select two categories, feldspatho-lithic (FL) and wackestone, which are frequently misclassified, for our verification process. Furthermore, we utilize category activation heat map visualization to gain insight into the influence of model predictions on recognition judgments. Fig. 11 presents the visualization and heat maps of the recognition results across these models.

In Fig. 11, the first and third rows are FL and wackestone limestone samples, respectively. The second and fourth rows correspond to the heat maps generated for each model. Despite the similar features of the components F, L, and Q, which can be challenging to distinguish based on content alone, RockNet demonstrates superior performance over the other five models in recognizing their distinct shapes, colors, and pleochroism.

### 5.6. Performance on multi-view and heterogeneous images

We further conduct comparison experiments to assess RockNet's performance on multi-view and heterogeneous images. To evaluate the efficiency of feature extraction, we categorize and label the debris grains in the four XPL images as Monocrystalline Quartz (Qm), Polycrystalline Quartz (Qp), and Chert (Cht). Subsequently, we examine the focus areas highlighted by RockNet using rectangles to delineate Qm, Qp, and Cht. Notably, PPL and XPL rock images exhibit distinctive features, even when captured from identical viewpoints. Sandstone classification is based on the relative proportions of its three main components: Q, F, and L. Fig. 12 illustrates the visualization and heat maps of recognition results.

As shown in Fig. 12, the eight heat maps correspond to four distinct views, with each featuring one PPL and one XPL image. The regions within the blue or red rectangles signify the key features of the ground truth or predictions made by RockNet, respectively. Areas enclosed by black rectangles indicate key regions that RockNet has overlooked. By examining the XPL activation heat maps for views 2 and 3, as shown in sub-figures (b) and (c), it is evident that RockNet concentrates on the particles Qm, Qp, and Cht, which are delineated by red rectangles. In contrast, the PPL activation heat map reveals that RockNet also attends to extraneous features, in addition to the key features marked by red rectangles. In sub-figure (d), we observe that RockNet omits two key features, indicated by black rectangles. Nonetheless, these features have been successfully captured by the XPL image of view 2. The diverse and complementary features hidden within the multi-view and heterogeneous images are effectively identified by RockNet, which

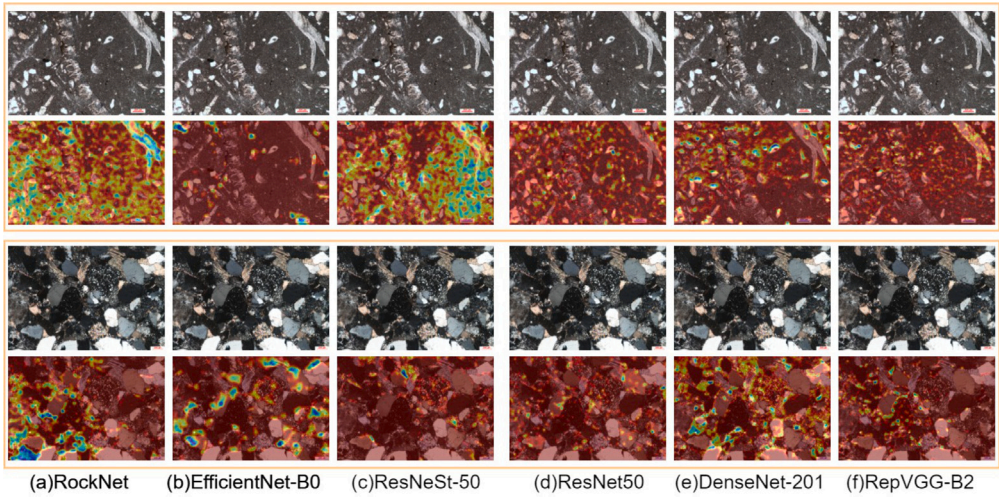



**Fig. 11.** Recognition heat maps: comparative analysis of RockNet against five leading models-original images of FL and wackestone (rows 1 & 3) with corresponding heat maps (rows 2 & 4).

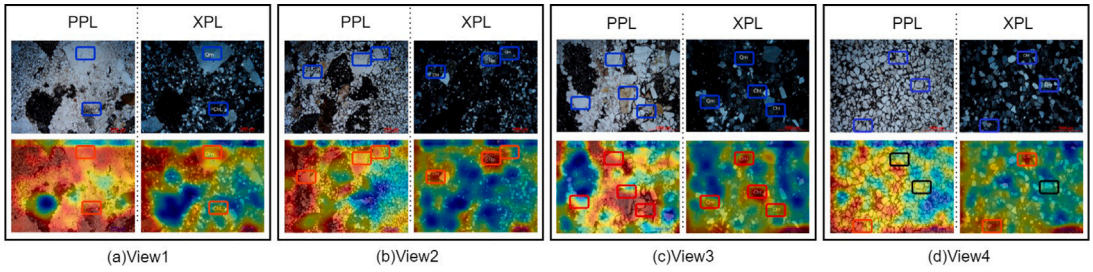



**Fig. 12.** Visualization and heat maps for quartzose sandstone: a multi-view and heterogeneous image analysis.

**Table 4**
Performance gains from RockNet strategies: a comparative analysis of category-balanced mini-batch generator, progressive training, and global–local feature augmentation.

| Method | Experimental result (%) | | | |
|---|---|---|---|---|
| RockNet baseline | ✓ | ✓ | ✓ | ✓ |
| Category-balanced mini-batch generator | | ✓ | ✓ | ✓ |
| Progressive training | | | ✓ | ✓ |
| Global-local feature augmentation | | | | ✓ |
| Accuracy | 87.1 | 88.3 | 89.1 | **90.1** |
| Precision | 84.3 | 85.6 | 85.5 | **89.0** |
| Recall | 80.9 | 82.9 | 81.7 | **85.8** |
| F1 score | 81.0 | 83.6 | 82.5 | **87.4** |
| Specificity | 99.1 | 99.2 | 99.3 | **99.3** |

accurately discerns key features from both PPL and XPL images across various views.

### 5.7. Ablation study

#### 5.7.1. Impact of category-aware augmentation and progressive training

We perform ablation experiments to assess the efficiency of the proposed category-aware augmentation and progressive training. Our design includes key strategies: global–local feature augmentation, progressive training, and a category-balanced mini-batch generator. We measure the impact of each strategy on the accuracy, precision, recall, F1 score, and specificity. Table 4 details the enhancements achieved through the strategies.

As presented in Table 4, the accuracy of the RockNet baseline is 87.1%. After applying the category-balanced mini-batch generator, progressive training, and global–local feature augmentation, the accuracy improves to 88.3%, 89.1%, and 90.1%, respectively. These improvements represent gains of 1.2%, 0.8%, and 1.0%, respectively, verifying the effectiveness of our category-aware augmentation and progressive training strategies.

The recognition results on a floatstone and an arenaceous are visualized in Fig. 13. The network improvement strategy effectively boosts the model's classification performance. Specifically, the addition of a category-balanced mini-batch generator helps to reduce the emphasis on background features by appropriately adjusting the network weights. After undergoing progressive training, the model effectively reduces the impact of background features in images, allowing RockNet to focus more accurately on small-sized particles. After applying global–local feature enhancement, the corresponding heat map indicates that local fine-grained features become more prominent and uniform. RockNet increases the emphasis on small-grained features, narrows the focus area, and enhances the recognition rate for small targets. In summary, enhancing the network architecture significantly improves the lithology recognition capabilities of RockNet. Moreover, each modification contributes to performance improvement in a unique way.

#### 5.7.2. Impact of loss contribution factors

In the proposed RockNet model, $\lambda$ is a factor representing the contribution of global features to the loss. In other words, $(1-\lambda)$ represents the weight of local features when calculating the loss. To evaluate its impact on the classification performance of RockNet and determine the optimal value, nine extended experiments are conducted. Table 5 shows the difference in RockNet recognition accuracy under various $\lambda$ settings.

Table 5 shows that RockNet exhibits strong classification capabilities when $\lambda$ is set to 0.4 and 0.7. Notably, at $\lambda = 0.4$, RockNet achieves the highest values for accuracy, recall, and F1 score, outperforming other settings considered. This configuration is especially advantageous for positively identifying rock samples, making it well-suited for practical rock-type classification tasks. Consequently, we set $\lambda$ to the default value of 0.4 in RockNet.

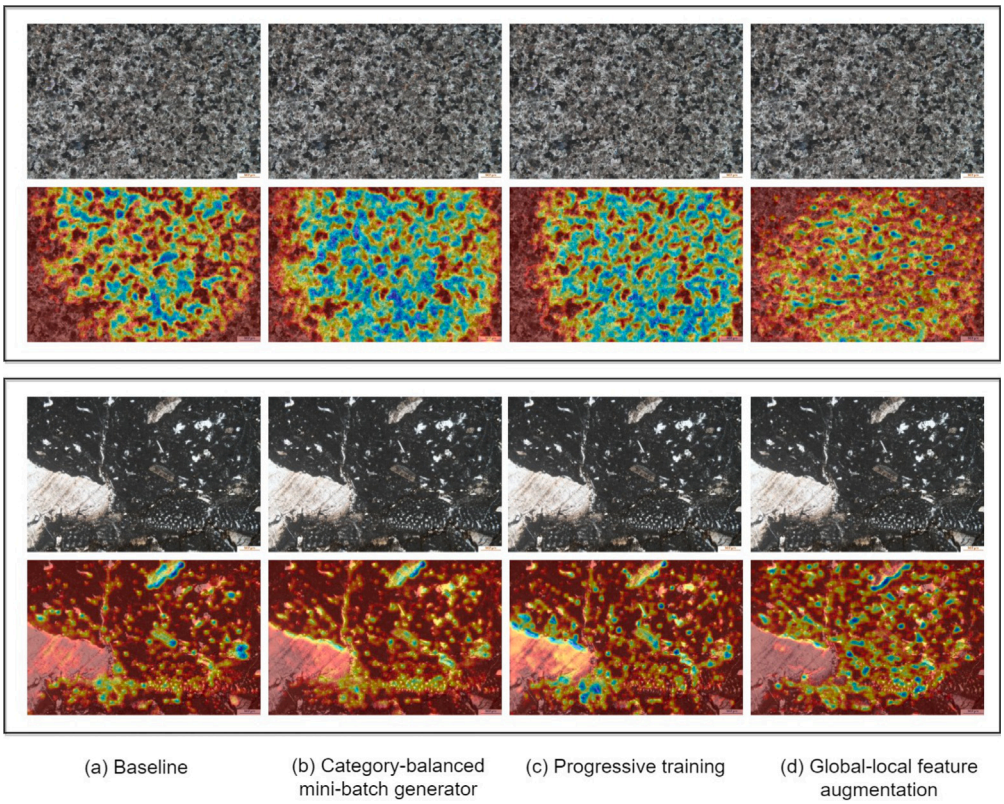



**Fig. 13.** RockNet strategy analysis: original floatstone and arenaceous rock images (rows 1 & 3) vs. corresponding heat maps (rows 2 & 4).

**Table 5**
Recognition accuracy under different $\lambda$ settings.

| $\lambda$ | Accuracy (%) | Precision (%) | Recall (%) | F1 score (%) | Specificity (%) |
|---|---|---|---|---|---|
| 0.1 | 89.8 | 88.8 | 85.6 | 87.2 | 99.3 |
| 0.2 | 89.2 | 86.6 | 83.2 | 84.9 | 99.3 |
| 0.3 | 89.7 | 87.5 | 83.8 | 85.6 | 99.3 |
| 0.4 | 90.1 | **89.0** | **85.8** | **87.4** | 99.3 |
| 0.5 | 89.3 | 87.2 | 82.4 | 84.7 | 99.3 |
| 0.6 | 89.8 | 87.6 | 84.2 | 85.9 | 99.3 |
| 0.7 | **90.2** | 88.8 | **85.8** | 87.3 | **99.4** |
| 0.8 | 88.3 | 85.0 | 82.0 | 83.5 | 99.2 |
| 0.9 | 88.5 | 86.0 | 82.7 | 84.3 | 99.2 |

## 6. Conclusion

Lithology identification is crucial for geological mapping and exploration. This paper proposes a new method named RockNet, which exploits the advantages of CNN to encode multi-view and heterogeneous features, thereby improving the accuracy of lithology identification. Comprehensive experiments on open-source datasets confirm the effectiveness of RockNet. Due to the rarity of certain rocks in real scenarios, training data is often insufficient. In future work, we will explore ways to improve the generalization capabilities of RockNet. Furthermore, with the advent of large models, rock image classification has made significant progress. Combining these large models with knowledge distillation provides a promising approach to rock lithology identification.

## CRediT authorship contribution statement

**Xiangyuan Zhu:** Writing – review & editing, Writing – original draft, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Mincan Li:** Writing – review & editing, Validation, Methodology, Funding acquisition, Formal analysis. **Zhiming Lan:** Visualization, Validation, Software, Methodology, Data curation. **Jianguo Chen:** Writing – review & editing, Writing – original draft, Supervision, Funding acquisition, Formal analysis. **Zerui Li:** Visualization, Validation, Software, Data curation. **Keqin Li:** Writing – review & editing, Methodology, Formal analysis.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mincan Li reports financial support was provided by National Natural Science Foundation of China. Jianguo Chen reports financial support was provided by National Natural Science Foundation of China. Jianguo Chen reports financial support was provided by Natural Science Foundation of Guangdong Province of China. Mincan Li reports financial support was provided by Natural Science Foundation of Hunan Province of China. Xiangyuan Zhu reports financial support was provided by Innovative Research Team of the Zhaoqing Big Data Engineering Technology Center. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors would like to thank Dr. Chih-Kuo Yeh for his guidance on model parameter configuration. This work was partially funded by the National Natural Science Foundation of China [Grant Nos. 62372486, 62206091], the Natural Science Foundation of Guangdong Province of China [Grant No. 2023A1515011179], the Natural Science Foundation of Hunan Province of China [Grant No. 2023JJ40166], and the Innovative Research Team of the Zhaoqing Big Data Engineering Technology Center.

## Data availability

The data and code are available at https://github.com/ZQU-BD/RockNet.

## References


[1] Z. Xu, W. Ma, P. Lin, H. Shi, D. Pan, T. Liu, Deep learning of rock images for intelligent lithology identification, Comput. Geosci. 154 (2021) 104799.

[2] Y. Liu, Z. Zhang, X. Liu, L. Wang, X. Xia, Deep learning-based image classification for online multi-coal and multi-class sorting, Comput. Geosci. 157 (2021) 104922.

[3] S. Dong, J. Hao, L. Zeng, X. Yang, L. Wang, C. Ji, Z. Zhong, S. Chen, K. Fu, A deep learning object detection method for fracture identification using conventional well logs, IEEE Trans. Geosci. Remote Sens. 62 (2024) 1–16.

[4] Y. Liu, W. Zhu, Y. Han, Enhancing texture feature for mineral classification in tight sandstone rock thin-section images using super-resolution techniques, Geoenergy Sci. Eng. 237 (2024) 212776.

[5] L. Zeng, W. Ren, L. Shan, Attention-based bidirectional gated recurrent unit neural networks for well logs prediction and lithology identification, Neurocomputing 414 (2020) 153–171.

[6] J. Chen, J. Yi, A. Chen, Z. Jin, EFCOMFF-Net: A multiscale feature fusion architecture with enhanced feature correlation for remote sensing image scene classification, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–17.

[7] J. Yang, Z. Kang, Z. Yang, J. Xie, B. Xue, J. Yang, J. Tao, A laboratory open-set Martian rock classification method based on spectral signatures, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–15.

[8] L. Zhang, J. Chen, J. Chen, Z. Wen, X. Zhou, LDD-Net: Lightweight printed circuit board defect detection network fusing multi-scale features, Eng. Appl. Artif. Intell. 129 (2024) 107628.

[9] Y. Bai, M. Liu, C. Yao, C. Lin, Y. Zhao, MSPNet: Multi-stage progressive network for image denoising, Neurocomputing 517 (2023) 71–80.

[10] X. Zhang, F. Yang, M. Chang, X. Qin, MG-MVSNet: Multiple granularities feature fusion network for multi-view stereo, Neurocomputing 528 (2023) 35–47.

[11] H. Liu, Y.-L. Ren, X. Li, Y.-X. Hu, J.-P. Wu, B. Li, L. Luo, Z. Tao, X. Liu, J. Liang, Y.-Y. Zhang, X.-Y. An, W.-K. Fang, Rock thin-section analysis and identification based on artificial intelligent technique, Pet. Sci. 19 (4) (2022) 1605–1621.

[12] A. Koeshidayatullah, M. Morsilli, D.J. Lehrmann, K. Al-Ramadan, J.L. Payne, Fully automated carbonate petrography using deep convolutional neural networks, Mar. Pet. Geol. 122 (2020) 104687.

[13] H. Ma, G. Han, L. Peng, L. Zhu, J. Shu, Rock thin sections identification based on improved squeeze-and-excitation networks model, Comput. Geosci. 152 (2021) 104780.

[14] H.L. Dawson, O. Dubrule, C.M. John, Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification, Comput. Geosci. 171 (2023) 105284.

[15] Y. Liu, Z. Zhang, X. Liu, L. Wang, X. Xia, Performance evaluation of a deep learning based wet coal image classification, Miner. Eng. 171 (2021) 107126.

[16] W. Zhou, H. Wang, Z. Wan, Ore image classification based on improved CNN, Comput. Electr. Eng. 99 (2022) 107819.

[17] X. Liu, H. Wang, H. Jing, A. Shao, L. Wang, Research on intelligent identification of rock types based on faster R-CNN method, IEEE Access 8 (2020) 21804–21812.

[18] R. Pires de Lima, D. Duarte, C. Nicholson, R. Slatt, K.J. Marfurt, Petrographic microfacies classification with deep convolutional neural networks, Comput. Geosci. 142 (2020) 104481.

[19] R. Chaganti, V. Ravi, T.D. Pham, A multi-view feature fusion approach for effective malware classification using deep learning, J. Inform. Secur. Appl. 72 (2023) 103402.

[20] X. Jia, X.-Y. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, D. Yue, Semi-supervised multi-view deep discriminant representation learning, IEEE Trans. Pattern Anal. Mach. Intell. 43 (7) (2021) 2496–2509.

[21] N. Zeng, P. Wu, Y. Zhang, H. Li, J. Mao, Z. Wang, DPMSN: A dual-pathway multiscale network for image forgery detection, IEEE Trans. Ind. Inform. 20 (5) (2024) 7665–7674.

[22] Y. Luo, Q. Huang, L. Liu, Classification of tumor in one single ultrasound image via a novel multi-view learning strategy, Pattern Recognit. 143 (2023) 109776.

[23] R. Du, J. Xie, Z. Ma, D. Chang, Y.-Z. Song, J. Guo, Progressive learning of category-consistent multi-granularity features for fine-grained visual classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (12) (2022) 9521–9535.

[24] B. Huang, Z. Wang, G. Wang, K. Jiang, Z. Han, T. Lu, C. Liang, PLFace: Progressive learning for face recognition with mask bias, Pattern Recognit. 135 (2023) 109142.

[25] H. Song, L. Chen, Y. Cui, Q. Li, Q. Wang, J. Fan, J. Yang, L. Zhang, Denoising of MR and CT images using cascaded multi-supervision convolutional neural networks with progressive training, Neurocomputing 469 (2022) 354–365.

[26] L. Hu, Z. Wang, H. Li, P. Wu, J. Mao, N. Zeng, $\ell$-DARTS: Light-weight differentiable architecture search with robustness enhancement strategy, Knowl.-Based Syst. 288 (2024) 111466.

[27] J. Peng, Y. Zeng, Y. Yang, L. Yu, T. Xu, Discussion on classification and naming scheme of fine-grained sedimentary rocks, Pet. Explor. Dev. 49 (1) (2022) 121–132.

[28] A.F. Embry, J.E. Klovan, A late devonian reef tract on northeastern banks Island, NWT, Bull. Can. Pet. Geol. 19 (4) (1971) 730–781.

[29] E. Garzanti, From static to dynamic provenance analysis sedimentary petrology upgraded, Sediment. Geol. 336 (2016) 3–13.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[31] X. Hu, W. Lai, Y. Xu, S. Zhang, X. Dong, Standards for digital micrograph of the sedimentary rocks, China Sci. Data 5 (3) (2020).

[32] R.J. Dunham, Classification of carbonate rocks according to depositional Texture1, in: Classification of Carbonate Rocks a Symposium, American Association of Petroleum Geologists, 1962, pp. 108–121.

[33] M. Tan, Q.V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, 2019, pp. 6105–6114.

[34] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, A. Smola, ResNeSt: Split-attention networks, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2022, pp. 2735–2745.

[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.

[36] S. Xie, R. Girshick, P. Doll r, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 5987–5995.

[37] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2261–2269.

[38] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepVGG: Making VGG-style ConvNets great again, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 13728–13737.

[39] I. Radosavovic, R.P. Kosaraju, R. Girshick, K. He, P. Doll r, Designing network design spaces, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 10425–10433.

[40] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 6848–6856.

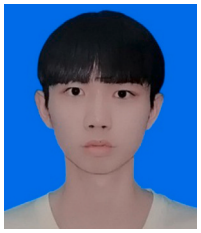
[41] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 11966–11976.

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16 × 16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, 2021, pp. 1–22.

[43] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 9992–10002.


**Xiangyuan Zhu** received her Ph.D. degree in computer science and technology from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2014. She is currently an associate professor at the School of Computer Science and Software, Zhaoqing University, Zhaoqing, China. Her major research interests include parallel computing, machine learning, knowledge distillation, and applications of artificial intelligence technology.

**Mincan Li** received the Ph.D. degree in computer science and technology from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2021. She is currently a postdoctoral researcher in the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. Her research interests include multiagent systems, many-objective optimization, and machine learning.



**Zhiming Lan** was born in Maoming, Guangdong, China, in 2001. He received his B.S. degree in data science and big data technology from the School of Computer Science and Software, Zhaoqing University, Zhaoqing, China, in 2024. His research interests include machine learning, big data analysis, applications of deep learning in data mining.

**Jianguo Chen** received his Ph.D. degree from the College of Computer Science and Electronic Engineering at Hunan University, China. He was a visiting Ph.D. student at the University of Illinois at Chicago from 2017 to 2018. He is currently an Associate Professor and one of the Hundred Academic Talents in the School of Software Engineering of Sun Yat-sen University (SYSU), China. Before joining SYSU, he was a postdoc at the University of Toronto in Canada and a research scientist at the A*STAR in Singapore. His major research interests include high-performance artificial intelligence, federated learning, and distributed computing. He has published more than 60 research papers in international conferences and journals such as IEEE-TII, IEEE-TITS, IEEE-TPDS, IEEEE-TKDE, IEEE/ACM-TCBB, and ACM-TIST. He is currently serving as an Associate Editor in the International Journal of Embedded Systems and Journal of Current Scientific Research.

**Zerui Li** was born in Shantou, Guangdong, China, in 2002. He received his B.S. degree in data science and big data technology from the School of Computer Science and Software, Zhaoqing University, Zhaoqing, China, in 2024. His main research topics include computer vision, big data mining, and applications of deep learning.

**Keqin Li** received a B.S. degree in computer science from Tsinghua University in 1985 and a Ph.D. degree in computer science from the University of Houston in 1990. He is a SUNY Distinguished Professor with the State University of New York and a National Distinguished Professor with Hunan University (China). He has authored or co-authored more than 1000 journal articles, book chapters, and refereed conference papers. He received several best paper awards from international conferences including PDPTA-1996, NAECON-1997, IPDPS, 2000, ISPA-2016, NPC, 2019, ISPA-2019, and CPSCom-2022. He holds nearly 75 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis *Who's Who in Science and Engineering*, *Who's Who in America*, *Who's Who in the World*, and *Who's Who in American Education* for over twenty consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department at the University of Houston in 2018. He received the IEEE TCCLD *Research Impact Award* from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC *Research Innovation Award* from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 *Technological Innovation Award* (Academic) in 2023. He is a Member of the SUNY Distinguished Academy. He is an AAAS Fellow, an IEEE Fellow, an AAIA Fellow, and an ACIS Founding Fellow. He is an Academician Member and Fellow of the International Artificial Intelligence Industry Alliance. He is a Member of Academia Europaea (Academician of the Academy of Europe).