Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Accurate Recognition of Drill Core Images with an Improved ResNet Model

Yong Pu[a], Zhihao Shao[b], Ziyuan Xu[b,*], Chuanhu Xiong[a] and Yonghua Chen[a]

[a] *Guangzhou Metro Design & Research Institute Co., Ltd., Guangzhou, China*

[b] *Institute of Intelligence Science and Technology, School of Computer Science and Software Engineering, Hohai University, Nanjing, China*

ARTICLE INFO

Article History:
Submitted 8.19.2024
Revised 8.31.2024
Second Revision 10.31.2024
Accepted 11.15.2024

Keywords:
Drill core image recognition
Self-calibration
Three-dimensional attention
MAM pooling

ABSTRACT

Drill core image recognition is an extremely critical aspect of geological exploration. However, the research on drill core image recognition faces challenges such as subtle discrepancies between different lithological categories, complex backgrounds, unclear recognition subjects, and lack of datasets. Currently, the existing recognition methods for geological images struggle to accurately classify drill core images. To address these issues, this paper con- structs a drill core image dataset called Drill Core Dataset (DCD), and proposes an image recognition model called MSL-ResNet50, which combines self-calibration residual module and three-dimensional attention. By introducing a self-calibration module with a larger receptive field, the model captures higher-level semantic information and enhances the network's feature transformation capability. Furthermore, it integrates the SimAM attention mechanism to mitigate interference in context information, giving higher attention to key areas. Additionally, the MAM pooling strategy is utilized to alleviate the loss of image features and maintain consistency in feature distribution of feature maps before and after pooling. Extensive ablation and validation experiments were conducted on the DCD dataset. The experimental results demonstrate that the proposed model, MSL-ResNet50, significantly outperforms the related benchmark models in drill core image recognition. Specifically, it achieves a recognition accuracy of 99.3%, an improvement of 4.1% over the original model, validating the effectiveness of the proposed model for core image recognition tasks.

1. Introduction

Image recognition is a key technology in computer vision to automatically identify and analyze objects and features in digital images. Drill core image recognition is a process specifically applied to geological exploration and petroleum engineering, aiming to automatically analyze core images to extract geological information. Traditional methods in drill core recognition rely on the experiences and technical capabilities of field surveyors, which is time-consuming, labor-intensive and highly subjective. Furthermore, differences in professional proficiency would lead to uneven accuracy of drill core identification results. Nowadays, automatic core image recognition can be achieved through deep learning.

Early lithology identification utilized the principal component analysis (PCA) to reduce the dimensionality of the original correlated lithology parameters [1, 2, 3]. Recently, deep learning has been applied to strata identification. Cheng proposed a particle size analysis method for rock images based on convolutional neural networks [4]. Zhang et al. accomplished the identification of three types of rocks by building a deep convolutional neural network model based on Inception-v3 [5]. Liu et al. proposed an optimal method for automatic identification of thin argillaceous interlayers, providing reference for the efficient development of oil sand projects [6]. Alférez et al. employed the convolutional neural network (CNN) model developed by TensorFlow to classify granite rocks [7]. Xu et al. proposed the Faster R-CNN architecture for rock image prediction, based on the ResNet structure and retaining the detailed information of the original image through residual learning [8]. Chen et al. established a classification framework for tunnel face rock structure based on the Inception-

[*] Corresponding author
Email address: hideon27@hhu.edu.cn

DOI reference number: 10.18293/JVLC2024N1-059

ResNet-V2 convolutional neural network [9]. Zhang et al. presented an improved Branch Module structure based on the AlexNet network to promote recognition accuracy and reduce the number of model parameters [10]. Bharadiya et al. combined a CNN model for image classification with learning representation to tackle the shortcomings of traditional feature selection methods [11]. Baraboshkin et al. applied CNN to the field of rock classification and suggested that GoogLeNet and ResNet are architectures with preferable performance [12]. Zhang et al. developed an intelligent system for identifying continental shale lithology using data balancing and deep learning, with the EfficientNet model performing best and providing technical support for evaluating continental shale reservoirs [13]. Wang et al. constructed an improved lightweight MobileViT model to analyze rock slice images covering most common lithology, addressing issues of unbalanced lithology datasets and numerous identification model parameters [14].

Despite remarkable progress in lithology recognition research, some challenges remain in this field. First, the images sampled at different times and locations would exhibit significant variations due to environmental influences, which places high demands on the generalization ability of the model. Second, since many rocks are highly similar in appearance and composition, the lithology of rocks presents the issue of large intra-class differences and small inter-class differences, making accurate identification difficult. Third, in dealing with diverse samples acquired from complex geological conditions, existing methods have difficulty in accurately separating and identifying each lithology in strata with mixed lithologies. This could be especially true when the lithology data has a highly unbalanced distribution.

To address the above challenges, on the basis of ResNet [13], a deep network that has been proven by the above-mentioned existing work to be effective in tackling lithology identification, this paper proposes an improved deep network model MSL-ResNet50 dedicated to the identification of drill core images, by optimizing the residual structure, pooling strategy and activation function of the original model. The main contributions of the paper are summarized as follows:

1. A dataset for drill core images, called Drill Core Dataset (DCD), is constructed. It includes 3070 drill core images, which are sampled under various environments, weather conditions, shooting angles, lighting, and depths.

2. A residual network that combines SimAM and self-calibrated convolution is constructed. The latent space is utilized to calibrate the original space to obtain more surrounding context information, including the contour and texture of the strata. The three-dimensional attention mechanism SimAM is employed to assist the latent space to extract information in a targeted manner, thus eliminating the interference in context information.

3. The MAM pooling strategy is introduced. The mean of each pooling window is compared with the standard deviation and mean of the entire pooling domain to select the appropriate pooling method, so that the feature map after pooling maintains the same feature distribution as the original image, and meanwhile obtains subtle features and significant features, and reduces the information loss caused by the pooling operation, thus improving the ability to distinguish between similar drill cores.

4. Extensive experiments on the data set DCD are conducted. The experimental results demonstrate that the proposed model significantly outperforms the existing models in lithology recognition. In particular, the accuracy of the improved model reaches 99.3%, which is 4.1% higher than the original model.

The remainder of the paper is organized as follows. Section 2 provides the overall architecture of the proposed model, Section 3 presents the self-calibrating convolution module with three-dimensional attention mechanism, Section 4 introduces the MAM pooling method, Section 5 conducts experiments and result analysis, and finally Section 6 concludes the paper.

2. Model Architecture

The architecture of the proposed model MSL-ResNet50 is shown in Figure 1. There are three fundamental components appearing in the network: CBL, MCB and CB, where CBL stands for Convolution, Batch normalization and LeakyReLU, MCB stands for MAM pooling, Convolution and Batch normalization, and CB stands for Convolution and Batch normalization. It incorporates three crucial improvements on the original model ResNet50.

As an improvement to the conventional residual module, MSL-ResNet50 constructs a module called sscblock that combines the self-calibrated convolution and the SimAM attention mechanism. This module divides the input into two parts. One part is the same as that of ResNet50, performing a standard 3×3 convolution, and the other is divided into two paths. Path 1 obtains the context information of the surrounding area after downsampling as a latent space with large receptive field, whereas Path 2 extracts features in the original scale space. Then, the outputs from the two paths are added and convolved to obtain the calibrated feature map. Since the self-calibration strategy would inevitably bring in useless background information, the SimAM mechanism is introduced in the latent space to acquire the context information, and the cross-dimensional information obtained from the interaction between the three dimensions of space and channel can be leveraged to reduce the interference of background information.

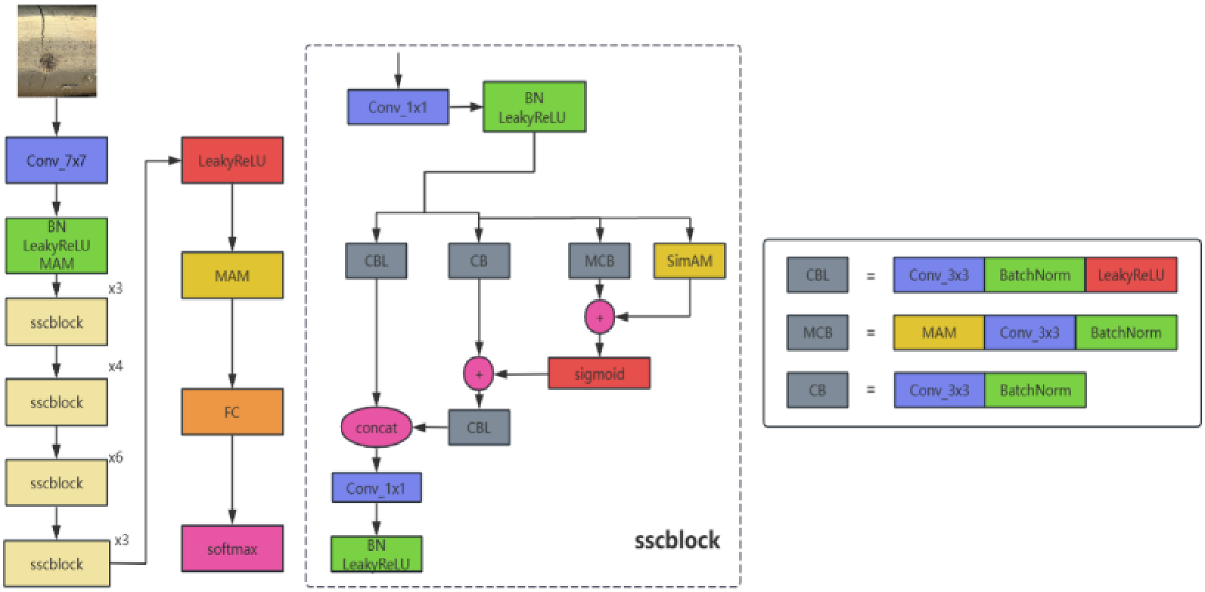

Figure 1: The structure of the proposed model MSL-ResNet50.

To tackle the issue that the original pooling method may cause information loss when reducing the model size, an improved strategy is proposed. Specifically, average pooling is not conducive to the extraction of edge features, and maximum pooling merely focuses on obvious features and ignores details and texture features, resulting in serious feature loss. Therefore, the MAM pooling method is employed to replace the original pooling, setting different pooling strategies in terms of the pooling window and the global situation. The flexible selection of pooling strategies can keep the style of the feature map before and after pooling consistent, and meanwhile enhances the model's perception of image details.

Additionally, the original activation function ReLU is replaced by LeakyReLU, to alleviate the issue of neuron “necrosis”. A comparative experiment has validated that LeakyReLU is the most beneficial activation function for enhancing the recognition performance of the model.

3. Self-Calibrated Convolution Module with 3D Attention Mechanism

Traditional convolution operations are performed by sliding a fixed-size convolution kernel over the input feature map. For a set of convolution kernels $K = [k_1, k_2, \dots, k_c]$, and input data $X = [x_1, x_2, \dots, x_c]$, the output data after convolution is denoted as $Y = [y_1, y_2, \dots, y_c]$. The output of the i-th channel can be expressed as $y_i = k_i * X = \sum_{j=1} k_i^j * x_j$.

This kind of convolution uses the same formula to calculate the output of each channel, and the final feature map is obtained by summing these outputs. Consequently, the features learned by the convolution kernels often lacks diversity, and the extracted feature map also lacks distinctiveness. In addition, the predefined convolution kernel size determines the receptive field of each spatial position, making it challenging for the network to effectively capture the high-level semantic information and generate optimal output results. The network constructed using such convolution layers exhibit clear drawbacks, such as insufficient receptive field and inadequate grasp of high-level semantic information.

Given that the feature transformation capability of a network is one of the critical factors affecting the recognition performance of convolutional neural networks, this section introduces a feature extraction method based on a self-calibrated residual network. This method leverages an internal message-passing mechanism to break away from the traditional use of small-size kernels for information fusion and extraction in both channel and spatial dimensions without adding additional learnable parameters.

However, when using self-calibrated convolutions to acquire surrounding context information, irrelevant background information would inevitably interfere with the network. Therefore, to assist the network in focusing attention on key beneficial areas, a self-calibrated convolution module with a three-dimensional attention mechanism is proposed. The module captures the critical information overlooked across three dimensions, avoiding the interference caused by long-range context information, and helps the model better understand the overall structure and context of the image.

3.1 Self-calibrated residual module

The structure of the Self-Calibrated Convolution (SCConv) is shown in Figure 2. It divides the convolution kernels of a specific layer into multiple parts, which are then fed into two different scale spaces for feature transformation. This allows for more effective capture of the rich context information surrounding each spatial position. In SCConv module, the self-calibration operation can be utilized to adaptively adjust the learning of long-distance spatial positions and the interaction between channels, thereby achieving the objective of expanding the receptive field of convolutions. The heterogeneous convolution communication means significantly expands the receptive field of each spatial position, thereby improving the perception ability of the network.

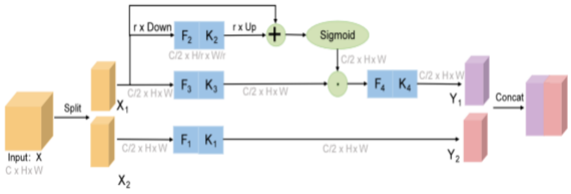

Figure 2: The structure of self-calibrated convolution module.

The specific calculation steps are as follows:

(1) The size of the input feature map is $C \times H \times W$. It is split into two parts, each with a size of $C/2 \times H \times W$, and sent into different paths to collect different types of context information.

(2) There are 4 convolution kernels, denoted as K_1, K_2, K_3, and K_4, each with dimensions $C/2 \times H \times W$.

(3) Processing the self-calibrated scale space:

For X_1, there are two parts: one entering the original space and the other entering the latent space. In the latent space, feature X_1 undergoes average pooling with a 4x downsampling:

$$T_1 = AvgPool_r(X_1) \tag{1}$$

Then, bilinear interpolation is used for upsampling, mapping the small-scale space back to the original feature space:

$$X_1' = Up\big(F_2(T_1)\big) = Up(T_1 * K_2) \tag{2}$$

where $*$ indicates the convolution operation. Then, a residual structure is constructed by addition and passed through the sigmoid activation function:

$$M_1 = \sigma(X_1 + X_1') \tag{3}$$

where σ is the sigmoid function, and X_1' serves as the residual to form the calibration weight. In the original space, X_1 is passed into the K_3 convolution kernel and calibrated using the features obtained from the latent space to produce Y_1:

$$\begin{cases} Y_1' = F_3(X_1) \cdot M_1 = (X_1 * K_3) \cdot M_1 \\ Y_1 = F_4(Y_1') = K_4 * Y_1' \end{cases} \tag{4}$$

where $F_3(X_1) = X_1 * K_3$, and $\cdot$ represents element-wise multiplication. $Y_1$is the final output after calibration.

(4) Processing the original space: perform a K_1 convolution operation on feature X_2 to extract feature Y_2;

(5) Concatenate the output features Y_1 and Y_2 from the two scale spaces to obtain the final output feature Y.

3.2 Three-dimensional attention mechanism

The essence of the attention mechanism lies in the process of to dynamically select information in the input image using different weights. In the current research on the combination of attention mechanisms and neural networks, CBAM (Convolutional Block Attention Module) is a representative attention mechanism module, which sequentially connects the channel attention module and the spatial attention module. Its structure is shown in Figure 3.

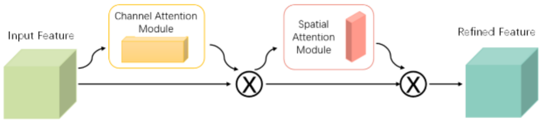

Figure 3: The CBAM structure.

The CBAM module can serially generate attention feature map information in the channel and space dimensions for the input feature map, and then achieve adaptive feature correction by multiplying it with the original input feature map. The structures of the channel attention and spatial attention are shown in Figure 4 and Figure 5, respectively.

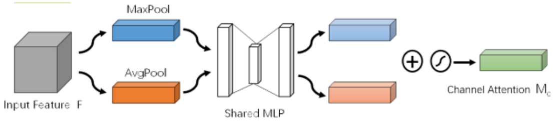

Figure 4: The channel attention structure.

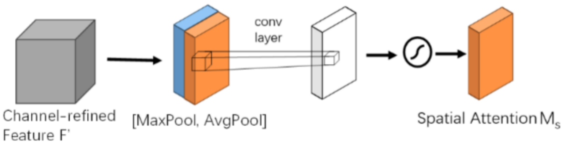

Figure 5: The spatial attention structure.

In terms of neuroscience theory, human spatial attention and channel attention often coexist and interact, jointly promoting visual processing. However, according to the above analysis, although the CBAM module takes into account the feature information of both spatial and channel dimensions, this serial approach makes the channel and spatial channels relatively independent, neglecting the interaction between them, which leads to the loss of important cross-dimensional information.

The SimAM mechanism can make up for the above shortcomings of the CBAM module. SimAM assigns a unique weight to each neuron without adding additional parameters, and treats the channel and spatial attention operations equally. This strategy generates three-dimensional weights that amplify the interaction features across all dimensions and reduce information diffusion.

Neuroscientific research indicates that if a neuron carries a large amount of information, its discharge pattern tends to significantly differ from that of other surrounding neurons. When these neurons are activated, they usually cause inhibition of surrounding neurons, a property known as spatial inhibition. In other words, in a neural network, neurons with spatial inhibition property should be given more attention and assigned higher weights compared to other neurons. The simplest way to identify such neurons is to measure the linear separability between neurons.

According to neuroscience theory, an energy function can be defined for each neuron to evaluate its importance, as shown in Equation 5:

$$e_t(w_t, b_t, y, x_i) = (y_t - t)^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}(y_o - \hat{x_i})^2 \quad (5)$$

where M represents the number of neurons in each channel, with $M = H \times W$; t denotes the target neuron and x_i denotes other neurons on the same channel as t; $\hat{t}$ indicates a linear transformation of t defined as $\hat{t} = w_t + b_t$, and $\hat{x_i}$ represents a linear transformation of x_i defined as $\hat{x_i} = w_t x_i + b_t$, where w_t and b_t are the weights and biases assigned during the linear change; and y_o and y_t represent binary labels.

In order to train the linear separability between the target neuron and other neurons in the same channel, binary labels are used, with $y_t = 1$ and $y_o = -1$, and a regularization term \$λ\$ is added to minimize Equation 5, yielding the final energy function:

$$\begin{aligned} e_t(w_t, b_t, y, x_i) = \frac{1}{M-1}\sum_{i=1}^{M-1}\Big[\big(-1 - (w_t x_i + b_t)\big)^2 \\ + \big(1 - (w_t + b_t)\big)^2\Big] + \lambda w_t^2 \quad (6) \end{aligned}$$

Theoretically, each channel has M neurons, resulting in M energy functions per channel. To reduce the computational overhead, the analytical solutions of w_t and b_t are derived:

$$\left\{ \begin{aligned} w_t &= -\frac{2(t - \mu_t)}{(t - \mu_t)^2 + 2\sigma_t^2 + 2\lambda} \\ b_t &= -\frac{1}{2}(t + \mu_t)w_t \\ \mu_t &= \frac{1}{M-1}\sum_{i=1}^{M-1} X_i \\ \sigma_t^2 &= \frac{1}{M-1}\sum_{i=1}^{M-1}(x_i - \mu_t)^2 \end{aligned} \right. \quad (7)$$

where μ_t represents the mean of all other neurons in the channel except for the target neuron, and σ represents the variance of all other neurons in the channel except for the target neuron. Since all neurons in each channel follow the same distribution, the minimum energy function for each position can be expressed as:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (8)$$

Equation 8 indicates that the linear separability between the target neuron and other neurons in the same channel is inversely proportional to its energy value. That is, when a neuron bears a lower energy, it is more distinctive from other neurons, and the importance of the neuron will be greater.

After determining the importance of neurons, the SimAM attention mechanism is used to enhance the features, resulting in a new feature map $\tilde{X}$, as shown in Equation 9. Here, E represents the set of all energy values in the input feature map, and "$\odot$" is the dot product. The Sigmoid function is applied to limit the impact of excessively large values in E.

$$\tilde{X} = sigmoid\left(\frac{1}{E}\right) \odot X \quad (9)$$

Building on the improved self-calibrated residual network discussed in Section 3.1, this section incorporates the non-parametric SimAM as the attention module within the network. This method not only enhances the fusion of the spatial and channel information without incurring additional computational costs, but also reduces the introduction of unimportant feature information when integrating surrounding context information through self-calibration. The structure of the improved self-calibration residual module with the SimAM attention is depicted in Figure 6.

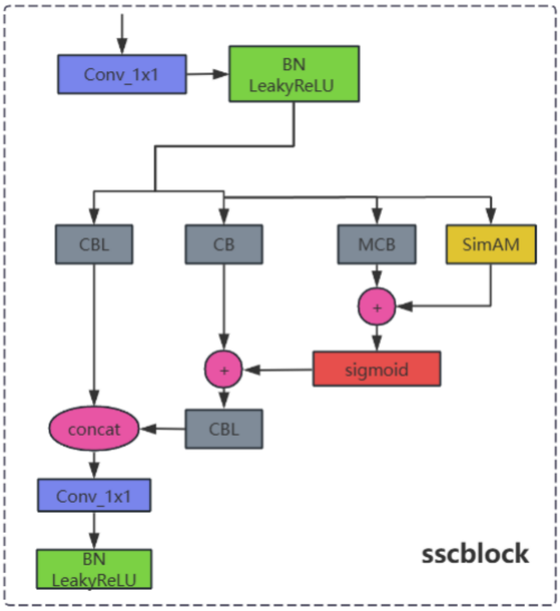

Figure 6: The structure of self-calibration residual with the SimAM attention.

4. MAM Pooling

The primary objective of pooling is to reduce the number of parameters of the model, minimize the interference of redundant information, and simultaneously retain critical feature information as much as possible, thereby maintaining the invariance of the feature map before and after pooling [13]. Currently, commonly used pooling methods include maximum pooling and average pooling. However, these two methods result in feature maps to be either close to its mean or close to its standard deviation after pooling, making it difficult to balance both.

When pooling feature maps, the approach that favors retaining the prominent features or overall style of the image leads to the incomplete understanding of the image. Retaining a part of the minimum values during pooling can preserve subtle features in the image and reduce the loss of overall features, thereby keeping the style and content features of the image before and after pooling unchanged. Additionally, the feature map after pooling shows clear contrasts, allowing for the exaction of sharp edges.

Therefore, it is necessary to design a dynamically selectable pooling method. To ensure that the feature map maintains consistent in content features and style before and after pooling, the mean $Avg(a_{ij})$ of each pooling window is compared with the weighted mean m_A and standard deviation s_A of the entire pooling region to determine whether to retain the maximum value, the average value, or the minimum value for each pooling window. The specific pooling process is expressed as follows:

$$\begin{cases} Max(a_{ij}) & Avg(a_{ij}) > m_A + \alpha s_A \\ Avg(a_{ij}) & m_A + \alpha s_A \geq Avg(a_{ij}) \geq m_A - \beta s_A \\ Min(a_{ij}) & Avg(a_{ij}) < m_A - \beta s_A \end{cases} \quad (10)$$

Here i and j represent the i-th row and j-th column of the feature map after pooling, A and B indicates the feature maps before and after pooling, respectively, α and β are adjustable parameters, and the pooling window size and step size are both k.

When $Avg(a_{ij})$ is greater than the weighted sum of m_A and s_A, it means that the data in the pooling window tends to be larger values. In this case, selecting the maximum pooling can better retain the prominent features. When $Avg(a_{ij})$ is less than the weighted difference between m_A and s_A, it means that the data in the pooling window tends to be smaller values. In this case, selecting the minimum pooling can better retain the subtle features. When $Avg(a_{ij})$ is between the weighted difference and the weighted sum of m_A and s_A , selecting the average pooling can better integrate the feature information within the window, thereby extracting a more comprehensive representation.

This strategy, at the cost of introducing a small amount of additional computation, can effectively choose the appropriate pooling method in terms of the overall situation of the data in the pooling window.

5. Experimental Results and Analysis

5.1 Datasets

We collected images of drill cores from different geological environments, including images taken at different time periods after drilling and from different sections, to comprehensively reflect the characteristics of cores, and constructed the Drill Core Dataset (DCD) accordingly. By retaining the influence of various factors such as different humidity levels, angles, and background interference, the dataset is made to be closely aligned with real-world application scenarios. The DCD dataset contains 15 representative rock and soil sub-layer categories, such as silt, silty soil, fine sand, medium-coarse sand, gravel sand, etc., totaling 3,070 images.

(1) Uneven data distribution

Figure 7 lists the number of samples for each category in the DCD dataset. As shown in the figure, the average number of samples per category is 264. The category with the most samples is plastic clay (4N-2), having 796 samples, while the category with the fewest samples is silty soil (2-1B), having 172 samples. Noticeably, the number of samples in the plastic clay (4N-2) category is significantly higher than in other categories. We applied data augmentation to all sample images and balanced the categories with large discrepancies in sample numbers.

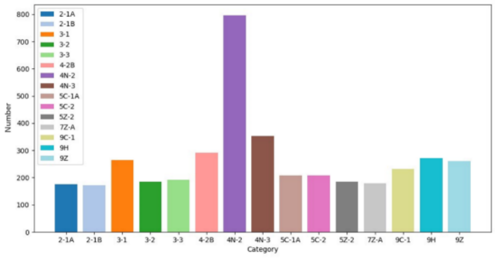

Figure 7: Number of samples in each category.

(2) Large intra-class differences, small inter-class differences

Through observation and comparison, it has been found that for the same type of drill core, different periods of photography exhibit varying humidity, texture, and shape characteristics. The same category of drill cores often includes two or more forms, with significant differences between them. Figure 8 illustrates silt clay with diverse colors and textures.

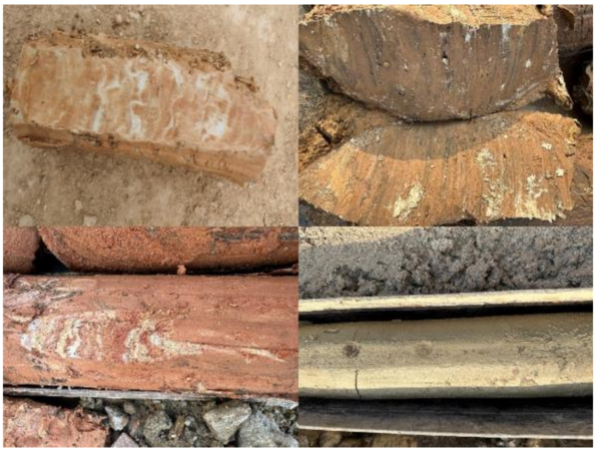

Figure 8: Examples of silty clay drill cores with diverse colors and textures.

For drill core images of different sublayers, there are often extremely similar forms that are difficult to distinguish. Figure 9 shows images of three categories: silt (2-1A), silty soil (4-2B), and silty clay (4N-2). Obviously, they have similar colors, identical shapes, and subtle inter-class differences.

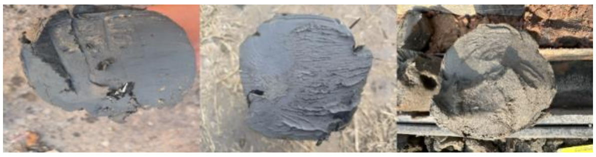

Figure 9: Examples of drill cores for different categories.

5.2 Experimental setup

The experimental settings and parameters are as follows: Adam is used as the optimizer, cross entropy loss function is utilized, batch size is 32, learning rate is 0.0001, number of epochs is 200, and the ratio of training set to validation set is 9:1. The preprocessing operations on the images before training include random horizontal flipping, random cropping, adding Gaussian noise, etc.

5.3 Experimental results and analysis

In this section, the results of extensive experiments are discussed from four perspectives. The first part is the set of comparative experiments of residual networks with different depths, the second part is the set of comparative experiments for different activation functions, the third part is the ablation study of the improved modules, and the fourth part is the set of comparative experiments of the proposed model and the baseline models.

(1) Comparative experiments of residual networks with different depths

The set of experiments was conducted using residual networks with different numbers of layers, and the experimental results are shown in Table 1.

Table 1: Experimental results of residual networks with different depths

Model	Top-1 Accuracy (%)
ResNet34	93.9
ResNet50	95.2
ResNet101	94.7

From Table 1, it can be seen that ResNet50 performs best on the DCD dataset. In addition, the 101-layer ResNet performs even worse than the 50-layer ResNet. This phenomenon is likely due to the small size of the DCD dataset, where a deeper network structure may lead to overfitting, resulting in performance degradation.

(2) Comparative experiments of activation functions

In this set of experiments, the activation function in the ResNet50 model were replaced with PReLU, ELU, GELU, and LeakyReLU, respectively, for comparison. The experimental results are shown in Table 2.

Table 2: Residual networks with different activation functions

Activation Function	Top-1 Accuracy (%)
ReLU	95.2
PReLU	95.6
ELU	94.3
GELU	94.0
LeakyReLU	96.4

The results show that the activation functions LeakyReLU and PReLU perform well on the ResNet50 model, enhancing the model’s recognition performance to some extent. By examining the computational efficiency and cost of both, LeakyReLU was chosen as the activation function due to its higher efficiency.

(3) Ablation study

In order to verify the effectiveness of each improved module, ablation experiments were conducted on the DCD dataset, including three models. Model 1 is a ResNet50 network using LeakyReLU as the activation function (L-ResNet50), Model 2 is formed by adding SCConv and SimAM modules to model 1 (SL-ResNet50), and Model 3 is formed by using MAM pooling on the basis of model 2 (MSL-ResNet50).The experimental results of the three models on the DCD dataset are shown in Table 3. Adding both the self-calibrated convolution SCConv and the three-dimensional attention mechanism SimAM, and the MAM pooling mechanism to the ResNet50 model with improved loss function, results in varying degrees of improvement in the network’s recognition accuracy, with increases of 1.9% and 1.0% respectively.

Table 3: Ablation study on the DCD dataset

Number	Model	Top-1 Accuracy (%)
1	L-ResNet50	96.4%
2	SL-ResNet50	98.3%
3	MSL-ResNet50	99.3%

Each of the three models was trained for 200 epochs, and the running results of validation accuracy of these models on the DCD dataset are shown in Figure 10.

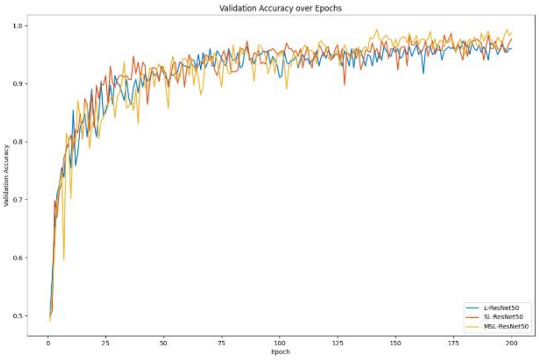

Figure 10: The curves of validation accuracy of the three ResNet50 models.

Table 4 shows the recognition results of the MSL-ResNet50 model on all the categories.

(4) Comparative experiments of related models

To further validate the classification performance of the proposed model for drill core images, we compared the proposed model MSL-ResNet with several representative benchmark models. The benchmark models include some mainstream image classification deep network models, such as AlexNet [15], GoogleNet [16], Vgg16 [17], Vision-Transformer [18], and the classification model ConvNet for plutonic rocks. In the experiments, all models used the same dataset partitioning, training epochs, and initial learning rate. Table 5 exhibits the classification results of the involved models and Figure 11 depicts the curves of validation accuracy of them. It can be observed that the MSL-ResNet50 proposed in this paper achieves significantly superior classification accuracy.

Table 4: The recognition precision and recall of MSL-ResNet50 for each category

Category	Precision	Recall
2-1A	0.941	0.941
2-1B	1.0	0.941
3-1	1.0	1.0
3-2	0.9	1.0
3-3	1.0	0.947
4-2B	0.954	0.954
4N-2	1.0	1.0
4N-3	1.0	1.0
5C-1A	1.0	1.0
5C-2	1.0	1.0
5Z-2	1.0	1.0
7Z-A	1.0	1.0
9C-1	1.0	1.0
9H	1.0	1.0
9Z	1.0	1.0

Table 5: Experimental results of involved models for comparison

Number	Model	Top-1 Accuracy (%)
1	AlexNet	93.4
2	GoogleNet	86.6
3	VGG16	87.3
4	Vision-Transformer	68.3
5	ConvNet	74.3
6	MSL-ResNet50	99.3

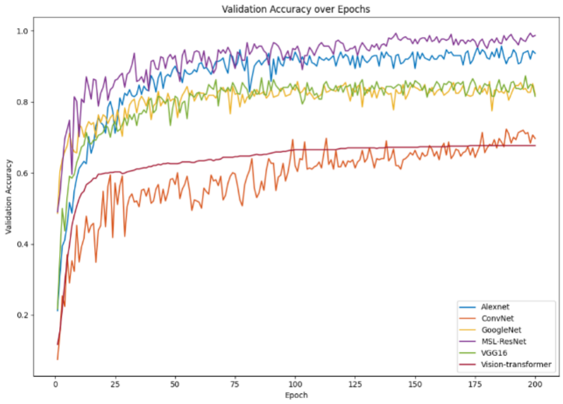

Figure 11: The curves of validation accuracy of the involved models for comparison.

(5) Grad-CAM feature visualization

Grad-CAM was employed to generate heatmaps for some selected samples, as shown in Figure 12. It can be observed that certain regions in the image samples receive high attention from the model, such as texture, color variations, and morphological features. These regions may contain features crucial for the model's classification decisions. Notably, these high-attention areas are concentrated in the actual core sections of the images rather than the background, indicating that the proposed model's focus on the target object' regions is appropriate and captures useful geological features.

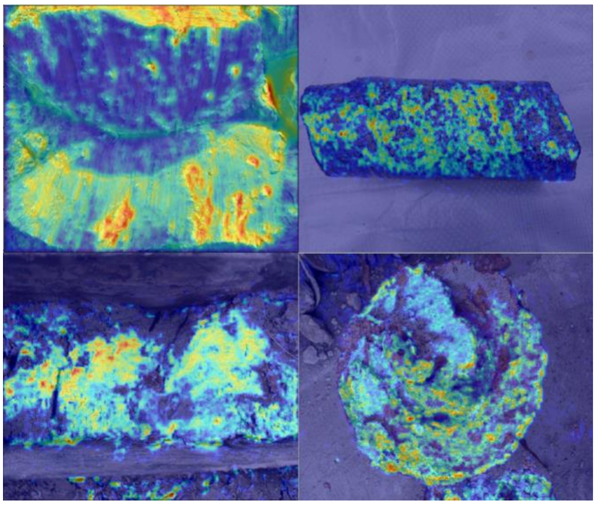

Figure 12: The curves of validation accuracy of the involved models for comparison.

6. Conclusion

References

[1] Y. Zhong and R. Li. Lithology identification method based on principal component analysis and least squares support vector machine. Well Logging Technology, 33(5):5, 2009.

[2] Y. Zhang and B. Pan. Application of principal component analysis-based som neural network in volcanic rock lithology identification. Well Logging Technology, 33(6):550– 554, 2009.

[3] A. Liu, L. Zuo, J. Li, et al. Application of principal com- ponent analysis in carbonate rock lithology identification — a case study of cambrian carbonate reservoir in YH area. Petroleum and Natural Gas Geology, 34(2):192–196, 2013.

[4] G. Cheng and W. Guo. Rock images classification by using deep convolution neural network. In Journal of Physics: Conference Series, volume 887, page 012089. IOP Publishing, 2017.

[5] Y. Zhang, M. Li, and S. Han. Lithology automatic recognition and classification method based on rock image deep learning. Acta Petrologica Sinica, 34(2):333–342, 2018.

[6] Y. Liu, J. Huang, Y. Yin, et al. Optimization and application of core image recognition algorithms for oil sand reservoirs. Fault-Block Oil & Gas Field, 27(4):464–468, 2020.

[7] G. H. Alferez, E. L. Vazquez, A. M. M. Ardila, et al. Automatic classification of plutonic rocks with deep learning. Applied Computing and Geosciences, 10:100061, 2021.

[8] Z. Xu, W. Ma, P. Lin, et al. Deep learning of rock images for intelligent lithology identification. Computers & Geo- sciences, 154:104799, 2021.

[9] J. Chen, T. Yang, D. Zhang, et al. Deep learning based classification of rock structure of tunnel face. Geoscience Frontiers, 12(1):395–404, 2021.

[10] B. Zhang. Research on neural network-based core image recognition algorithms. PhD thesis, Yangtze University, 2022.

[11] J. Bharadiya. Convolutional neural networks for image classification. International Journal of Innovative Science and Research Technology, 8(5):673–677, 2023.

[12] E. E. Baraboshkin, L. S. Ismailova, D. M. Orlov, et al. Deep convolutions for in-depth automated rock typing. Computers & Geosciences, 135:104330, 2020.

[13] Z. Zhang, J. Tang, B. Fan, et al. An intelligent lithology recognition system for continental shale by using digital coring images and convolutional neural networks. Geoenergy Science and Engineering, 239:212909, 2024.

[14] Q. Wang, J. Yang, F. C. Huo, et al. Lithology identification method of rock thin section images based on mobilevit. Geological Bulletin of China, 43(6):938–946, 2024.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, volume 25, 2012.

[16] C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.