


Article

Rock Thin Section Image Identification Based on Convolutional Neural Networks of Adaptive and Second-Order Pooling Methods

Zilong Zhou, Hang Yuan * and Xin Cai * 

School of Resources and Safety Engineering, Central South University, Changsha 410083, China

* Correspondence: 205511039@csu.edu.cn (H.Y.); xincal@csu.edu.cn (X.C.)

Abstract: In order to enhance the ability to represent rock feature information and finally improve the rock identification performance of convolution neural networks (CNN), a new pooling mode was proposed in this paper. According to whether the pooling object was the last convolution layer, it divided pooling layers into the sampling pooling layer and the classification pooling layer. The adaptive pooling method was used in the sampling pooling layer. The pooling kernels adaptively adjusted were designed for each feature map. The second-order pooling method was used by the classification pooling layer. The second-order feature information based on outer products was extracted from the feature pair. The changing process of the two methods in forward and back propagation was deduced. Then, they were embedded into CNN to build a rock thin section image identification model (ASOPCNN). The experiment was conducted on the image set containing 5998 rock thin section images of six rock types. The CNN models using max pooling, average pooling and stochastic pooling were set for comparison. In the results, the ASOPCNN has the highest identification accuracy of 89.08% on the test set. Its indexes are superior to the other three models in precision, recall, F1 score and AUC values. The results reveal that the adaptive and second-order pooling methods are more suitable for CNN model, and CNN based on them could be a reliable model for rock identification.



Citation: Zhou, Z.; Yuan, H.; Cai, X. Rock Thin Section Image Identification Based on Convolutional Neural Networks of Adaptive and Second-Order Pooling Methods. *Mathematics* **2023**, *11*, 1245. <https://doi.org/10.3390/math11051245>

Academic Editor: Fernando Simoes

Received: 8 February 2023

Revised: 25 February 2023

Accepted: 1 March 2023

Published: 4 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: rock; rock thin section image; rock identification; convolution neural networks; pooling layer

MSC: 68T07

1. Introduction

Rock identification is a basic and prerequisite work of geological engineering [1]. For instance, geologists need lithological type information to infer the history of regional geological evolution, judge types of deep mineral resources, as well as oil and gas resources, and invert reserve information of various resources [2,3]. Engineers need lithological type information to guide and design the construction of geotechnical engineering, such as mining and tunnelling [4,5]. Insufficient rock type information may lead to a series of engineering disasters, including landslide, collapse, and settlement [6,7]. Hence, it is necessary to study how to identify rock types accurately and quickly.

Many scholars have conducted lots of research on rock identification and put forward many methods, which can be summarized into the following four types: microscopic observation, experimental tests, statistics and learning, and deep learning [8].

Microscopic observation and experimental tests belong to manual identification methods. They observe rock characteristics through optical microscopes, and analyze compositions and structures of rocks and minerals with the help of X-ray Diffraction (XRD), Electron Microprobe Analysis (EMPA), etc. Zhang [9] introduced the identification principle and process of rock microscopic observation in combination with cases. In order to identify

acid volcanic rocks, Liu et al. [10] used X-ray Fluorescence Spectroscopy (XRF) to analyze the principal components of rocks and EMPA to check and show the minerals. Manual identification methods have achieved certain results; however, they are time-consuming, labor-intensive, costly, subjective, and greatly affected by professional levels of observers as well as professionalism of instruments. It is not applicable for rock identification of large-scale stratum in engineering.

The automatic identification methods based on rock features and machine learning have gradually gained the favor of scholars [11,12]. Chatterjee et al. [13] input color, shape and texture features extracted from rock images into SVM to identify rock types, and finally achieved an accuracy of 96.2%. Patel et al. [14] extracted nine color histogram features from rock images and input them into the probabilistic neural network. They successfully identified limestone types, with an error rate of less than 6%. Zhang et al. [15] used five machine learning models to identify rock and mineral images, and then selected three models with the best performance to stack. The stacked model effectively improved model performance. Machine learning methods can realize automatic identification of rock types. However, features required to classify still need to be selected subjectively by professionals. Feature types selected usually are few [8]. In addition, features of different lithology have different preferences, resulting in many problems in practical application.

In recent years, artificial intelligence (AI) methods and technologies have developed rapidly [16–20]. As a core method in the field of target detection, target identification and target segmentation, deep learning has been gradually applied to geological and rock engineering [21–26].

As a visual model of deep learning, convolutional neural networks (CNN) can automatically select the most suitable features to distinguish different type rocks. Rock type identification using CNN is usually based on rock thin section images. Rock thin sections are rock slices, which are observed and studied under the polarizing microscope. The rock slices are made from large rock samples through cutting, grinding and other operations, and they are about 60 mm × 60 mm in size and about 0.3 mm in thickness. Many scholars have performed lots of innovative research on rock thin section image identification. Polat et al. [27] used DenseNet121 and ResNet50 to identify six types of volcanic rock and tested the impact of four different optimizers on model accuracy. Alzubaidi et al. [28] used the architecture of ResNeXt-50 to identify the rock types of oil and gas reservoir logging core images, and the final accuracy reached 93.12%. In order to improve the identification accuracy, Liu et al. [29] built a mineral image identification model based on ResNet, embedding four visual attention blocks. Ma et al. [30] proposed the MaSE-ResNeXt model to enhance feature connectivity between different channels, and the identification accuracy on three kinds of rock thin section images finally reached 90.89%. Li et al. [8] researched the influence of three different optimization algorithms and two attenuation methods of learning rate on identification performance. Dos Anjos et al. [31] believed that the existing CNN needed to unify the input image size, which would lose the original image information inevitably. Their research work proposed a CNN based on a pyramid pooling layer. The image was down sampled according to the pyramid layering mode, which can process input images of all sizes. The final research showed that this method can improve the identification accuracy to a certain extent. Su et al. [32] believed that different shooting types of rock images had an impact on the final accuracy. Their research work inputted three types of rock images into three identical CNN, respectively, including the plane polarized light image, the cross polarized light image and the image after principal component analysis. The final rock type was determined through the maximum likelihood method based on the results of three CNN. The final accuracy reached 89.97%. Seo et al. [33] researched the impact of local images on the identification accuracy. They believed that the features in local areas in the rock image were more representative and definite. They proposed a model based on image segmentation. The large image was divided into several small parts, which were input into CNN in turn. The final lithology category was the one with the highest quantity of local identification results. Xu et al. [34] researched the impact of the fusion

of image and data features on rock identification. Their research work proposed a fusion identification method, which inputs image features and parts of lithological data into the full connection layer of CNN. The final results showed that this method can improve the identification accuracy. Zhang et al. [15] researched the effect of different classifiers on the performance of rock identification models. They identified and classified the rock image features extracted from CNN by using five classifiers: logical regression (LR), support vector machine (SVM), random forest (RF), k-nearest neighbors (KNN), multilayer perceptron (MLP) and gaussian naive Bayes (GNB). The final result showed that the classification performance of the logistic regression, support vector machine and multilayer perceptron was better than other methods. However, the above research still has some deficiencies. Current research mainly focuses on the function of convolution layers, full connection layers and optimization algorithms of CNN, ignoring the effect of pooling layers on the performance of rock identification. Current pooling layers only play the role of down sampling. The current pooling methods used are main max pooling and average pooling. They are static and can only extract less information about rock features. Max pooling is sensitive to mutation of pixels, so it can retain texture features of rocks, as well as shape and size features of minerals to the maximum extent [35]. Average pooling pays more attention to preserving overall feature information of input images. It can better preserve color features, background features and global combination features of mineral composition in location and content [29]. However, internal structures and mineral morphology of rocks contain dozens or even hundreds of features, and the features required to distinguish different rocks are distinctive. Static pooling methods cannot select the best pooling method according to different lithological characteristics. At the same time, current pooling methods can only obtain first-order feature information from a single feature map, ignoring the relationship between feature pairs. Adequate exploration of feature distributions is important for realizing the full potentials of CNN [36].

Therefore, the functions of pooling layers were divided into two types: down sampling and classification, and the adaptive and second-order pooling methods were designed and respectively applied in this paper. The adaptive pooling set pooling kernels for each feature map. The parameters in pooling kernels participated in the training process, and finally they were configured by error feedback. During the process of the second-order pooling, the second-order feature information based on outer products between feature pairs was extracted and finally input into a classifier. After deducing the changing process of forward and back propagation, the proposed pooling methods were embedded into CNN to build an identification model of rock thin section images. Another three models using traditional pooling methods were used for comparison. The performance was comprehensively evaluated with multiple indicators to provide a reliable model for rock identification.

2. Data Collection

The rock thin section images used in this paper were from the symposium on Microscopic Images of Rocks in the open-source database, China Scientific Data [37–42]. There were 3374 images collected from the open-source database. Six kinds of common rocks were selected for experiment, including wackestone, granite, schist, quartz sandstone, conglomerate and crystalline dolomite. The sites of rock samples and the production process of rock thin section images were shown in Figure 1a. As shown in Figure 1b, these rock samples came from geological drillings in different regions of China. Through the steps in Figure 1c, rock samples were cut, polished and stuck into rock thin sections, and then they were photographed under polarizing microscopes to obtain microscopic images. Wackestone is a kind of sedimentary rock. The matrix filling of it is plaster, and the particles are mainly bioclasts and cuttings of rocks. Granite is a kind of igneous rock with holocrystalline structures. Its main mineral compositions are quartz, feldspar, and a small amount of mica. Schist is a kind of metamorphic rock with lamellar structures. Its common mineral compositions include quartz, quartzite, and mica. Quartz sandstone is a kind of sedimentary rock with clastic structures. Its mineral composition is mainly quartz,

containing a small amount of feldspar. The cement between minerals is calcareous. The diameter of mineral particles is mostly between 0.2 mm and 2 mm. Conglomerate is a kind of sedimentary rock with clastic structures. Its mineral compositions are mainly quartz and feldspar and contains some cuttings of rocks. The cement between minerals is siliceous. The diameter of mineral particles is mostly between 2 mm and 5 mm. Crystalline dolomite is a kind of sedimentary rock with recrystallized structures. The crystalline minerals are mostly quartz, with diameters ranging from 0.005 mm to 0.03 mm. The cement between minerals is siliceous.

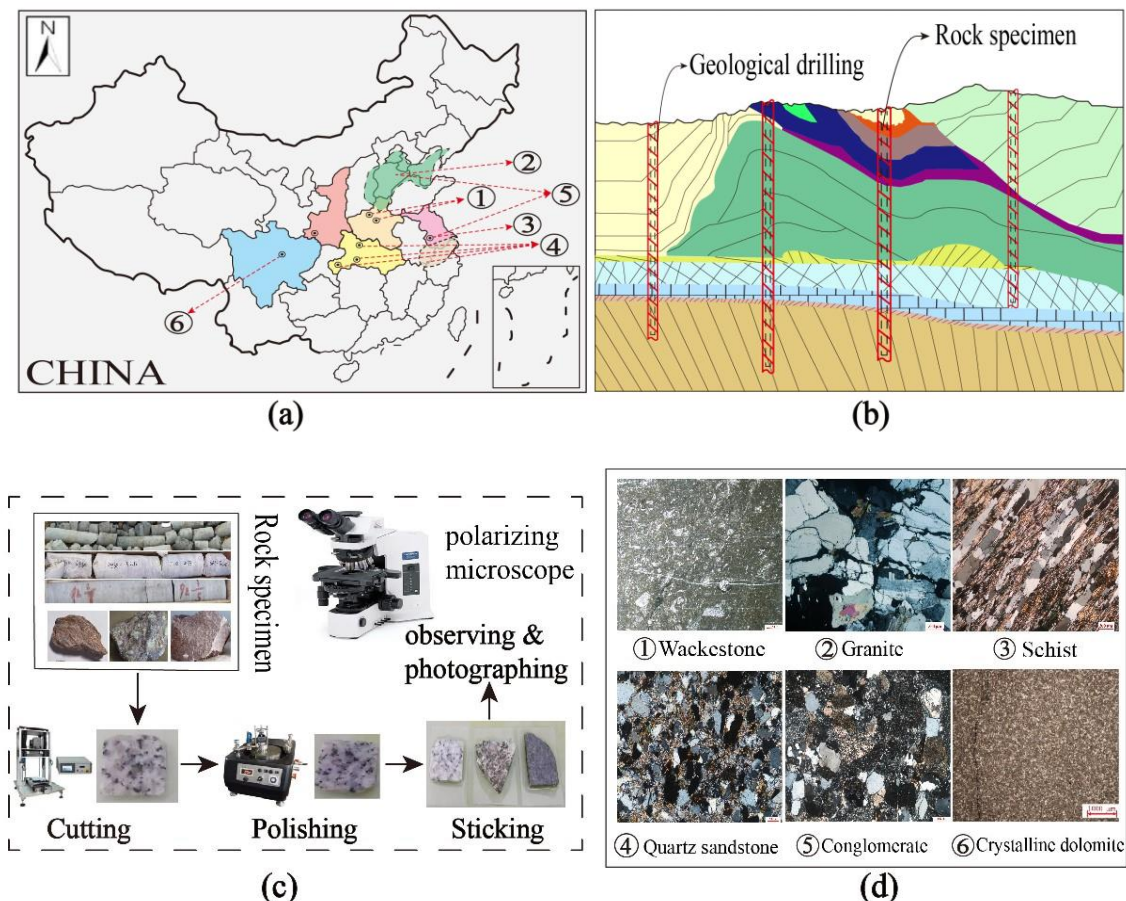


Figure 1. The sites of rock samples and the production process of rock thin section images; (a) Rock sample acquisition area; (b) Lithological map of different stratum; (c) Manufacturing process of rock thin sections; (d) Rock thin section image set.

To prevent over fitting in the training process and increase robustness of results, several commonly used image enhancement technologies were used to expand the image set, including image rotation, image flip, brightness change, noise addition and histogram equalization [30,33]. The image set contained 5998 images after enhancement. It was roughly divided into the training set, validation set and test set with the ratio of 7:2:1. The specific division details can be seen in Table 1. The pixels of the images in the original set were not uniform, while the CNN model required the input images to be consistent in size, so the bilinear interpolation algorithm was used to unify the image size to 227×227 pixels. The process of bilinear interpolation was shown in Figure 2. According to the coordinate defined in Figure 2, the pixel values of the interpolation points can be calculated by Equations (1) and (2); the pixel value of the target point can be calculated by Equation (3).

$$f(x, y_0) = \frac{x_1 - x}{x_1 - x_0} f(x_0, y_0) + \frac{x - x_0}{x_1 - x_0} f(x_1, y_0) \quad (1)$$

$$f(x, y_1) = \frac{x_1 - x}{x_1 - x_0} f(x_0, y_1) + \frac{x - x_0}{x_1 - x_0} f(x_1, y_1) \quad (2)$$

$$f(x, y) = \frac{y_1 - y}{y_1 - y_0} f(x, y_0) + \frac{y - y_0}{y_1 - y_0} f(x, y_1) \quad (3)$$

where $f(x, y)$ is the pixel value of the target point; $f(x, y_0)$, $f(x, y_1)$ are pixel values of the interpolation point; $f(x_0, y_0)$, $f(x_1, y_0)$, $f(x_0, y_1)$, $f(x_1, y_1)$ are pixel values of the original points.

Table 1. Rock thin section image set.

Rock Type	Training Set	Validation Set	Test Set	Total
Wackestone	636	180	103	919
Granite	732	216	92	1040
Schist	672	192	89	953
Quartz sandstone	756	216	102	1074
Conglomerate	768	216	104	1088
Crystalline dolomite	648	180	96	924
Total	4212	1200	586	5998

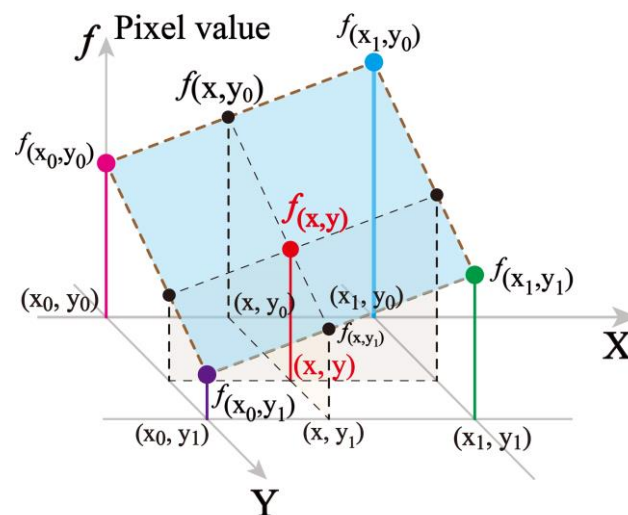


Figure 2. The calculation process of the bilinear interpolation algorithm.

3. Proposed Method

3.1. Basic CNN Model

The structure of CNN can be divided into two modules: feature extraction and feature classification, and its overall workflow can be divided into two processes: forward propagation and back propagation (parameters updating). Its basic structure was shown in Figure 3.

Images to be identified were input into the feature extraction module, which was stacked by multiple convolution layers and pooling layers. The module was used to extract the underlying features of images. Then, the extracted features were input into the feature classification module, which was composed of full connection layers and a softmax layer. Its function was to classify the input features. The final output was a vector, in which each element value represented the probability that the image belonged to the corresponding category. Through this process, a forward propagation was completed. Then the sum of the error between calculated results and real values was taken as the total loss of CNN. The partial derivatives of the total loss to parameters were taken as the error sensitivity. The error sensitivity of each layer was calculated by back propagation. Then the parameter values were optimized by optimizers. Through this process, a back propagation process was completed. The gradient descent method was usually used as the optimizer in CNN. Its calculation process was shown in Equation (4). The CNN model completed an iteration

through forward and backward propagation. The specific function of each layer was introduced below.

$$new(w) = old(w) - \eta \delta w \quad (4)$$

where δw was the error sensitivity of weight parameter w ; η was the learning rate.

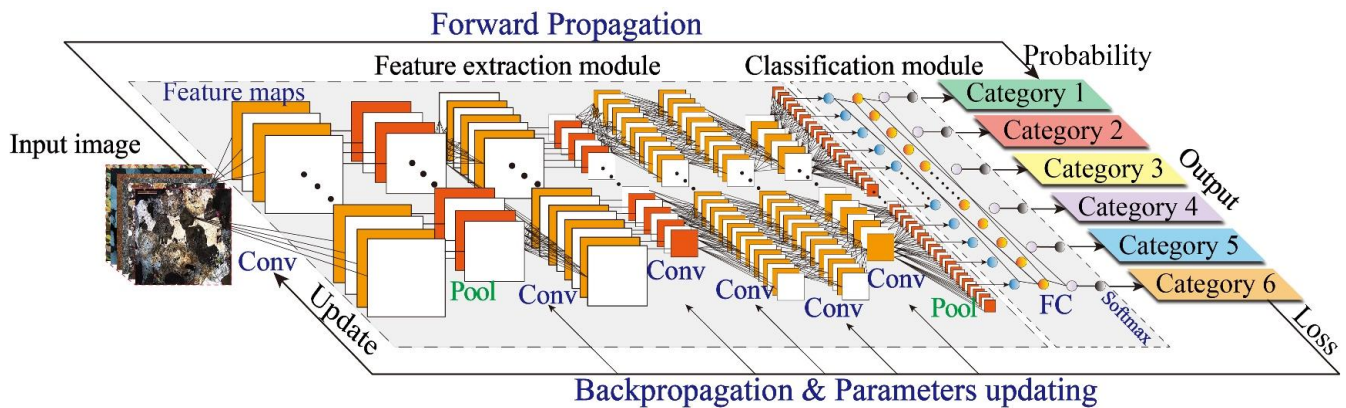


Figure 3. The structure and workflow of CNN model.

The function of convolution layers is to extract feature maps. Convolution kernels slide across the input images from left to right and top to bottom. At each sliding position, the sub-region elements and convolution kernels perform convolution operations as shown in Equation (5). The output results constitute feature maps.

$$X_j^l = f\left(\sum_{i \in M_j} X_i^{l-1} \otimes k_{ij}^l + b_j^l\right) \quad (5)$$

where X_j^l is the j th feature map of the l th layer, k_{ij}^l is the convolution kernel of the l th layer, \otimes is the convolution operation symbol, which represents the sum of the multiplication of corresponding position elements of two matrixes. The function of activation layers is to increase the nonlinearity of output and make multi-layer stacking meaningful. The commonly used activation functions include ReLU and Sigmoid. The function of pooling layers is to reduce the size of the feature maps. Pooling layers output results through a sliding window similar to convolution layers. Each sliding outputs a special value of the corresponding sub-region. The function of the full connection layers is to classify the input feature data. It can map the input features to the sample tag space and obtain the values of the sample belonging to each category. The calculation process is shown below.

$$X_j^l = f\left(\sum_i X_i^{l-1} \cdot W_i + b_j^l\right) \quad (6)$$

The function of the softmax layer is to convert the results from full connection layers into probabilities between 0 and 1. The calculation process is shown below.

$$p(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (7)$$

3.2. Rock Thin Section Image Identification Model

In this section, the pooling layers were first divided into the sampling pooling layer and the classification pooling layer according to whether the pooling object was the last convolution layer. Then, the adaptive pooling method was designed for the sampling pooling layer, and the second-order pooling method was designed for the classification pooling layer. Then based on the Alexnet [43] framework, these two pooling methods were

embedded to design a rock thin section image identification model. Alexnet is a classic CNN model, which won the championship in the IMAGENET-2012 competition. It was widely used in rock image identification because of its small model and fast computation [44], the adaptive pooling and second-order pooling methods were embedded to build a feature extractor, and then the feature extractor was combined with a classifier to construct a rock thin section image identification model (ASOPCNN). The overall structure of the model was shown in Figure 4. Rock thin section images with 227×227 pixels were input into the feature extractor, which included 5 convolution layers, 2 adaptive pooling layers and 1 s-order pooling layer. The detailed parameters of the convolution and pooling layers were shown in left bottom of Figure 4. The parameter table in the convolution layer successively represented the kernel size and number, the layer name, the moving step and padding value. The parameter table in the pooling layer successively represented the kernel size and number, the layer name and the pooling step. The second-order pooling layer specifically included five layers as shown in lower right corner of Figure 4. In order to speed up the training process, the 2nd, 4th and 5th convolution layers were set as the upper and lower parts, using two GPU for parallel computing. The activation function used the ReLU. The rock features extracted were input into the classifier. The classifier contained 3 full connection layers and 1 softmax layer. The numbers of neurons in the full connection layer were 4096, 4096, 6, respectively. The probability values belonging to each rock category were output as the results.

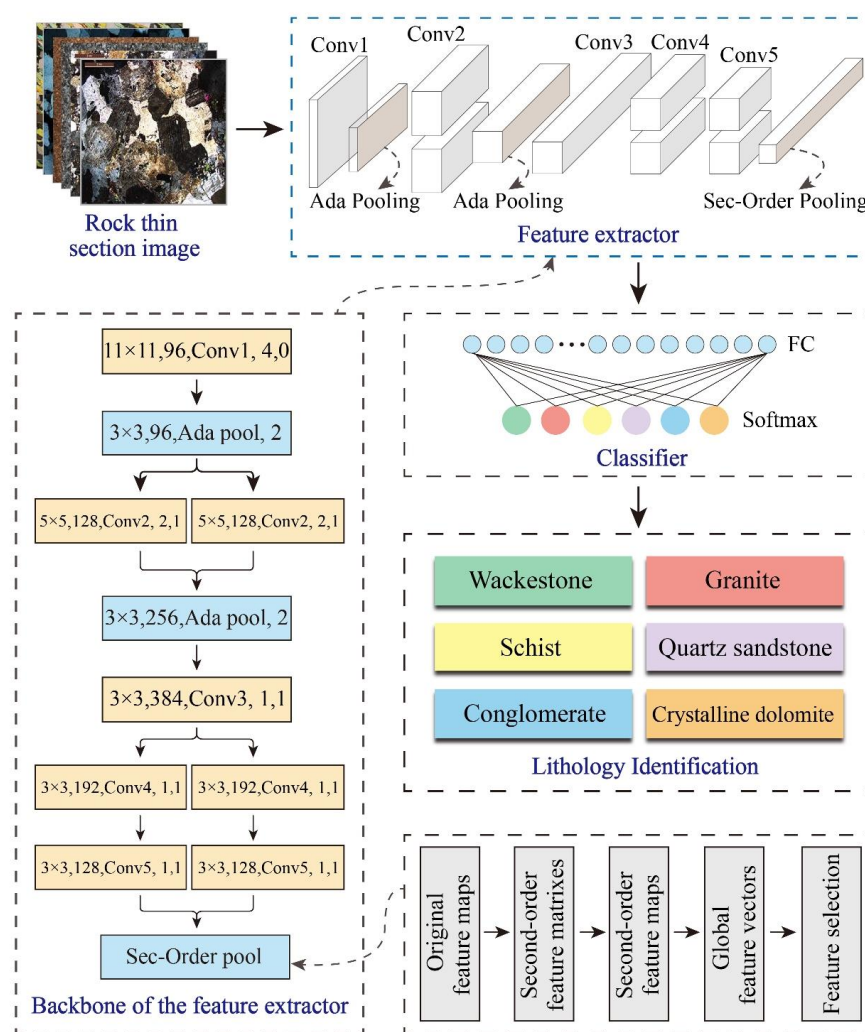


Figure 4. The structure of rock thin section image identification model (ASOPCNN).

3.3. Adaptive Pooling Method

The purpose of sampling pooling layers is to reduce the size of feature maps [45]. The traditional pooling methods are mainly max pooling, mean pooling and stochastic pooling. Their pooling processes were shown in Figure 5. With max pooling, the maximum value of each region was output. The average value of each region was output through average pooling. With stochastic pooling, the probability of each element in the region was calculated, and then the output was selected randomly according to the magnitude of probability [20]. However, pooling methods have certain selectivity in the process of representing rock features. Figure 6 shows the selectivity of two pooling methods. It can be seen in Figure 6a that texture features became more prominent after max pooling, while the number of textures decreased, and the structure becomes fuzzy after average pooling. It showed that max pooling had a strengthening effect on texture features, while average pooling had a restraining effect. Figure 6b shows the pooling process of wackestone. From the color histogram, we can see that color values of the original image were concentrated in the range of 50 to 200 but are relatively dispersed. After average pooling, the value range remained unchanged, and the aggregation degree was higher. It indicated that some disturbing color information was lost after average pooling. However, the color range changed after max pooling. It indicated that the original color feature had been lost, which also proved the inapplicability of max pooling to color features.

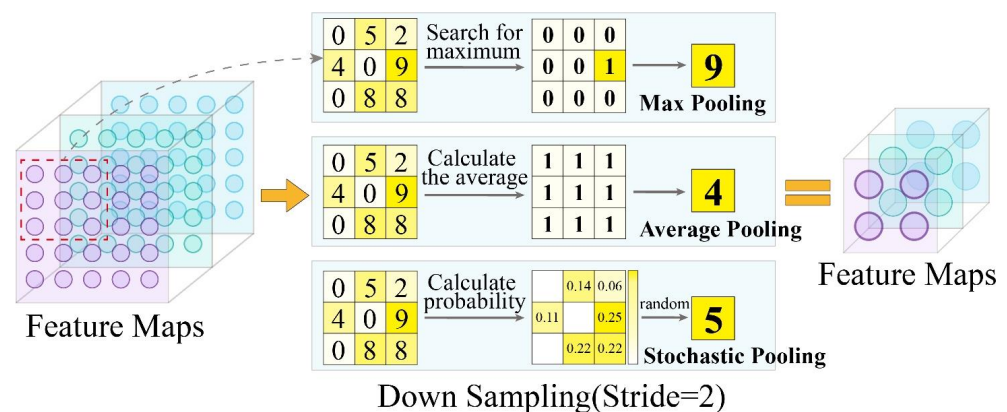


Figure 5. Process of traditional pooling methods, including max pooling, average pooling and stochastic pooling.

No matter what traditional pooling method is used, all feature maps use a single sampling way, which will inevitably cause loss of feature information and affect performance of CNN. An adaptive pooling method was proposed in this paper, and its pooling process was shown in Figure 7. For input feature maps, firstly, the pooling kernels were deployed for each feature map; each kernel was independent of each other. Then, initial values were set randomly between 0 and 1 for each kernel. Finally, convolutional results were calculated between the feature maps and the pooling kernels through Equation (8).

$$O = \frac{\sum_{i=1}^m \sum_{j=1}^n I_{ij} \cdot W_{ij}}{\sum_{i=1}^m \sum_{j=1}^n W_{ij}} \quad (8)$$

where O was the output value; I was the input value; W was the parameter of pooling kernels. The parameters were adaptively adjusted, and the process can be divided into the following four steps, as shown in the right side of Figure 7.

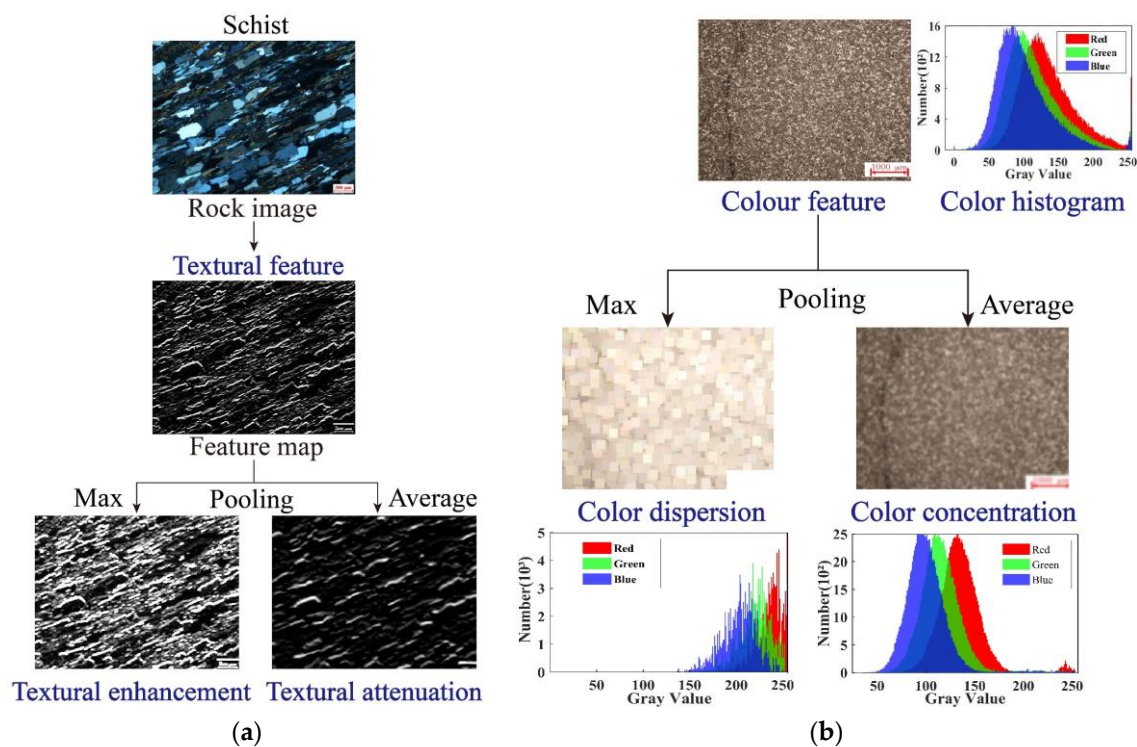


Figure 6. (a) The selectivity of textural features in the pooling process; (b) The selectivity of color features in the pooling process.

Step 1: Calculate total loss L . The total loss was the difference between real values and actual output values.

Step 2: Calculate partial derivatives. The partial derivatives of total loss to each parameter of pooling kernels were calculated layer-by-layer through back propagation.

Step 3: Update parameters. According to the partial derivatives and the learning rate, new parameters were calculated through the gradient descent method.

Step 4: Limit parameter boundary. If the calculated parameter was bigger than 1, then it was equal to 1; if the calculated parameter was smaller than 0, then it was equal to 0.

The adaptive pooling method affected the forward and back propagation process of CNN. Its forward propagation was the convolutional calculation of the target feature maps with the pooling kernels. The result feature maps can be calculated by Equation (9).

$$y_{i,j} = \sum_{m=1}^b \sum_{n=1}^b w_{m,n} \cdot x_{s \cdot (i-1) + m, s \cdot (j-1) + n} \quad (9)$$

where $x_{i,j}$ and $y_{i,j}$, respectively, represented the values in row i and column j of the target feature maps and the result feature maps; $w_{m,n}$ represented the value in row m and column n of pooling kernels; b represented the size of pooling kernels; s represented the step size.

The back propagation process needed to calculate $\delta x_{i,j}$ and $\delta w_{i,j}$ through Equations (10) and (11). $\delta x_{i,j}$ was the error sensitivity of target feature maps, which represented the partial derivative of the total loss to each element in the target feature maps. In the same way, $\delta w_{i,j}$ was the error sensitivity of pooling kernels, which was used to calculate error sensitivity of front layers.

$$\delta x_{i,j} = \sum_{m=1}^c \sum_{n=1}^c \delta y_{m,n} \cdot w_{i-s \cdot m + s, j-s \cdot n + s} \cdot f[i - s \cdot (m-1)] \cdot f[j - s \cdot (n-1)] \quad (10)$$

$$\delta w_{i,j} = \sum_{m=1}^c \sum_{n=1}^c \delta y_{m,n} \cdot x_{s \cdot (m-1) + i, s \cdot (n-1) + j} \quad (11)$$

where $\delta y_{m,n}$ was the error sensitivity of the result feature map; c was the size of the result feature maps. $f(x)$ was a judgement function, which equaled to 1 while $0 < x < b$ and equaled to 0 for else. The detailed derivation process of forward and back propagation can be seen in Appendix A.

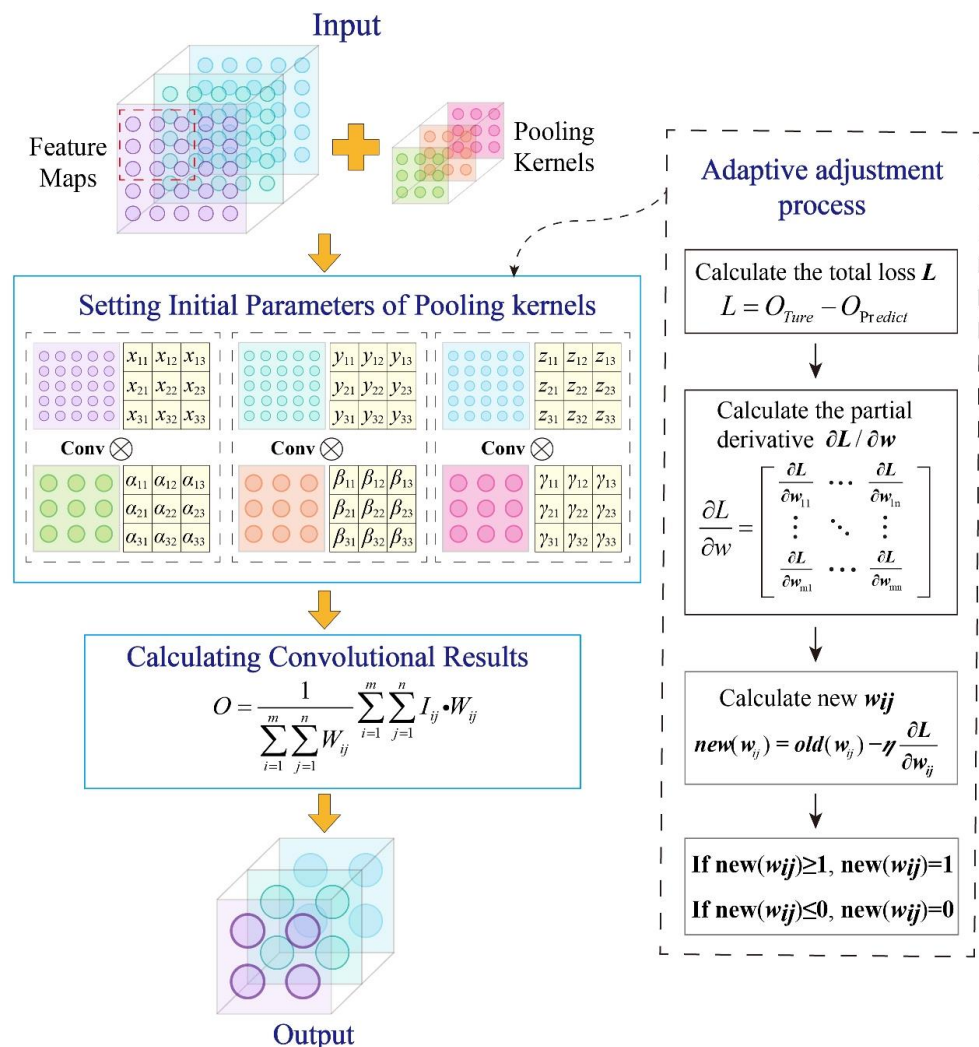


Figure 7. Process of the adaptive pooling method.

3.4. Second-Order Pooling Method

The classification pooling layer is connected behind the last convolution layer. Its purpose is to integrate feature maps into a compact global feature. Then the global feature is input into a full connection layer for classification. The process of integrating feature maps for traditional pooling methods was shown in Figure 8. Each feature map was expanded into a vector after down sampling, then all vectors were connected to form a global feature vector in turn. However, this method only counted first-order feature information. Because in the integration process, each feature map was independent and uncorrelated. As a result, the ability of feature representation was limited.

Carreira et al. [46] proposed a method based on outer products to extract second-order feature information. However, they only used it for image segmentation. Based on their work of them, the second-order pooling method was proposed in this paper by embedding the second-order information extraction process into CNN and deriving its forward and back propagation. The process of the second-order pooling was shown in Figure 9. The whole process can be divided into the following six steps.

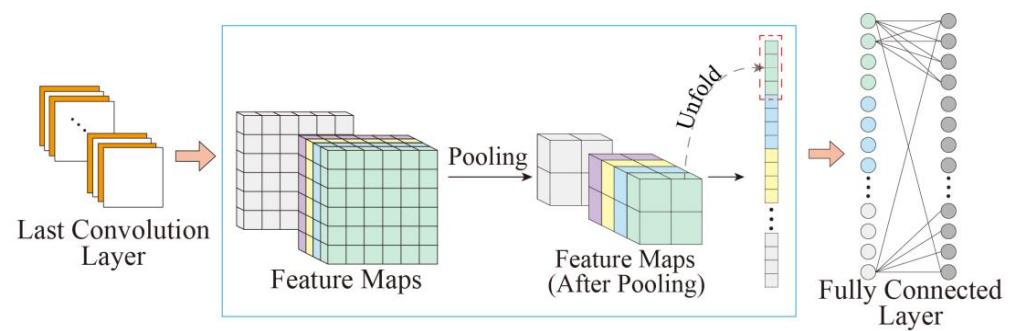


Figure 8. Process of integrating feature maps of traditional pooling methods.

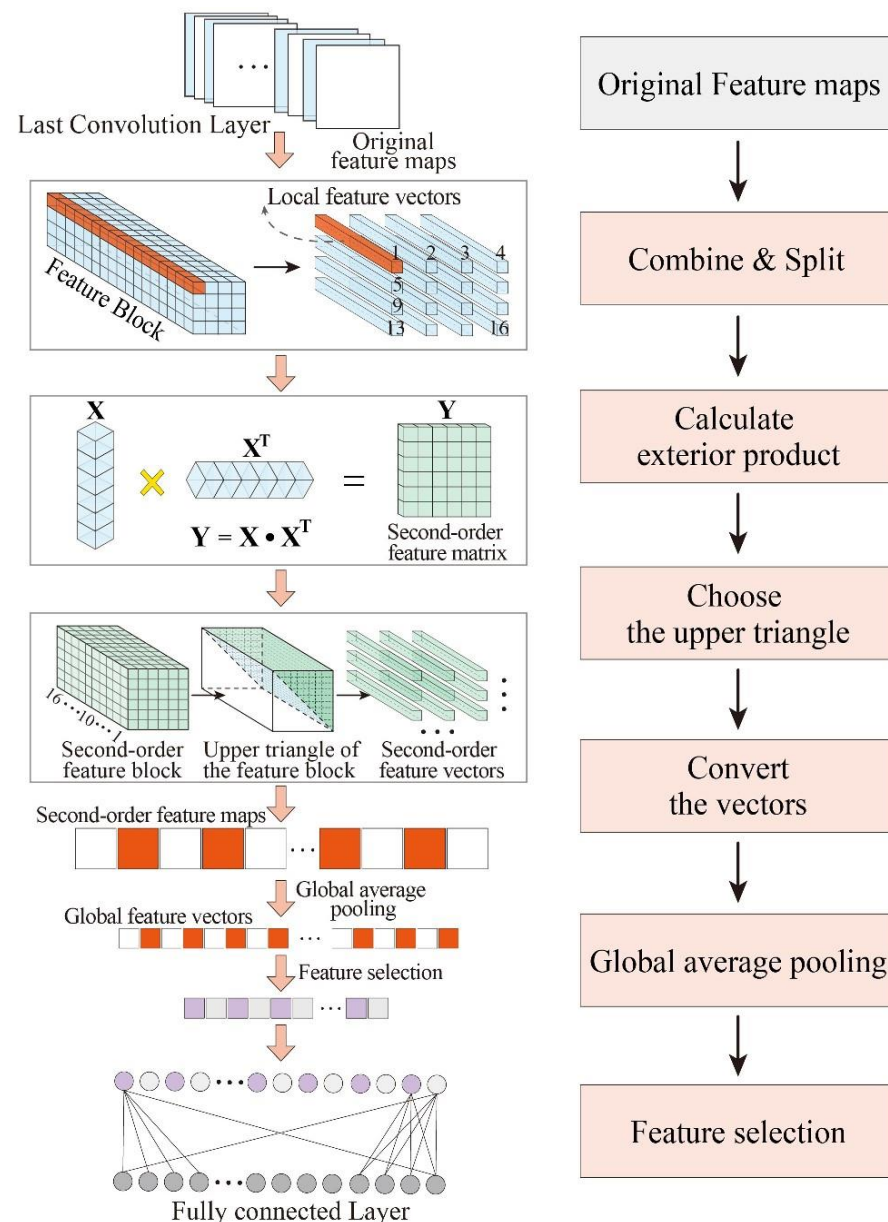


Figure 9. The process of the second-order pooling method.

Step 1: Combine and split. For feature maps of the last convolution layer, they were combined to form a feature block; then, the feature block was divided into local feature vectors in sequence from left to right and top to bottom according to the element position. The local feature vectors contained information of all feature maps at a single position.

Step 2: Calculate outer products. The local feature vectors were taken as the main body, then outer products of each local feature vector were calculated to obtain the second-order feature matrixes.

Step 3: Choose upper triangle part. All second-order feature matrixes were combined to form a second-order feature block, and then the block was divided into second-order feature vectors in sequence from left to right and top to bottom according to the element position. To avoid duplication, only the upper triangle part of the block was retained. Each second-order feature vector contained information of all positions in each original feature map.

Step 4: Convert vectors into matrixes. Each second-order feature vector was converted into a second-order feature map in sequence from left to right and top to bottom according to the size of the original feature map.

Step 5: Global average pooling. The second-order feature maps were down sampled to obtain the global feature vector through global average pooling.

Step 6: Feature selection. Because the positive excitation function, such as ReLU and Sigmoid functions, was generally used in the process of feature extraction, the numerical difference of elements can reflect the importance of feature information. The second-order pooling can further expand this difference. In order to reduce the number of parameters and speed up the training process, a part of feature information was selected in sequence from large to small according to the importance of feature information. The number of features as a super parameter needed to be preset.

The second-order pooling method affected forward and back propagation of CNN. In forward propagation process, the elements in the second-order feature matrixes, second-order feature maps and global feature vectors can be, respectively, obtained by Equations (12)–(14).

$$y_{i,j,k} = \sum_{m=1}^a \sum_{n=1}^a x_{m,n,i} \cdot x_{m,n,j} \cdot I[a(m-1) + n = k] \quad (12)$$

$$z_{\alpha,\beta,\gamma} = \sum_{k=1}^{a^2} \sum_{i=1}^b \sum_{j=1}^b y_{i,j,k} \cdot I[b(i-1) + j - \frac{i^2-i}{2} = \gamma] \cdot I[c(\alpha-1) + \beta = \gamma] \quad (13)$$

$$Z_p = \frac{\sum_{\alpha=1}^c \sum_{\beta=1}^c z_{\alpha,\beta,p}}{\alpha \cdot \beta} \quad (14)$$

where $x_{m,n,l}$ represented the element in row m and column n of the l th original feature map; $y_{i,j,k}$ represented the element in row i and column j of the k th second-order feature matrix; $z_{\alpha,\beta,\gamma}$ represented the element in row α and column β of the γ th second-order feature map; Z_p represented the value of the p th element in the global feature vector; a, b, c , respectively, represented the size of the original feature maps, the second-order feature matrixes and the second-order feature maps; I (condition) was a judgment function; if the condition was met, then $I = 1$, otherwise $I = 0$.

In the back propagation process, since there were no parameters to be updated, only elements of error sensitivity were required to calculate. The error sensitivity of the second-order feature maps, the second-order feature matrixes and the original feature maps can be, respectively, obtained by Equations (15)–(17).

$$\delta z_{\alpha,\beta,\gamma} = \frac{\delta Z_\gamma}{\alpha \cdot \beta} \quad (15)$$

$$\delta y_{i,j,k} = \sum_{\alpha=1}^c \sum_{\beta=1}^c \delta z_{\alpha,\beta,b \cdot (i-1) + j - (i^2-i)/2} \cdot I[b(\alpha-1) + \beta = k] \quad (16)$$

$$\delta x_{m,n,l} = \sum_{i=1}^b \sum_{j=1}^b \delta y_{i,j,s_2 \cdot (m-1) + n} \cdot x_{m,n,j} \cdot I(i=l) \cdot [I(i=m) \cdot I(j=n) + 1] \quad (17)$$

where δZ_γ represented the error sensitivity of the global feature vector; $\delta z_{\alpha,\beta,\gamma}$ represented the error sensitivity of the second-order feature maps; $\delta y_{i,j,k}$ represented the error sensitivity of the second-order feature matrixes; $\delta x_{m,n,l}$ represented the error sensitivity of the original feature maps. The detailed derivation process of the forward and back propagation can be seen in Appendix B.

3.5. Evaluation Metrics

In order to effectively evaluate classification performance of CNN models, several indicators were used, including accuracy rate (ACC), precision rate (PRE), recall rate (REC), F1 score (F1), confusion matrix and the receiver operating characteristic curve (ROC) [47–49]. The confusion matrix was shown in Figure 10, through which the identification effect of CNN models can be observed intuitively. In the confusion matrix, each row is the prediction labels of samples, and each column is the real labels of samples. P (Positive) represents the prediction label is positive; N (Negative) represents the prediction label is negative; T (True) represents the sample is predicted correctly; F (False) represents the sample is predicted wrongly.

Confusion Matrix		True Value	
		Positive	Negative
Prediction Value	Positive	TP	FN
	Negative	FP	TN

Metric	Definition
ACC	The proportion of correct classification samples
PRE	The proportion of correct classification in samples which were predicted to be positive
REC	The proportion of correct classification in samples which were positive realistically
F ₁	The harmonic mean value of PRE and REC

Figure 10. Schematic of the confusion matrix.

The accuracy rate (ACC) represents the proportion of correct classification, which can be calculated by Equation (18). The precision rate (PRE) represents the proportion of correct classification in labels which were predicted to be positive, which can be calculated by Equation (19). The recall rate (REC) refers to the proportion of correct classification in labels which were realistically positive, which can be calculated by Equation (20). F1 score is the harmonic mean value of PRE and REC, which can be calculated by Equation (21).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

$$PRE = \frac{TP}{TP + FP} \quad (19)$$

$$REC = \frac{TP}{TP + FN} \quad (20)$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (21)$$

The receiver operating characteristic curve (ROC) can show the performance of models under different classification thresholds. The ROC curve uses the false positive rate (FPR) as the abscissa and the true rate (TPR) as the ordinate, which can be calculated by Equations (22) and (23), respectively. AUC is the area under the curve, and large AUC value indicates good performance of models. When $AUC \leq 0.5$, prediction result is less effective than random guess, and the model has no predictive worth.

$$FPR = \frac{FP}{TN + FP} \quad (22)$$

$$TPR = \frac{TP}{TP + FN} \quad (23)$$

4. Results and Discussions

The program of the proposed model was written based on Matlab language and the deep learning library in Matlab 2021b. It was 32 k in size. The experimental process was carried out on the server with a 64-core CPU, a 192 G RAM and the Linux operating system, which belonged to the Super Cloud Computing Center in Beijing, China. The average cost of per image during training was 1.257 s. In order to verify the effectiveness of the adaptive and second-order pooling methods, the original model of Alexnet, which used max pooling, was used for comparative verification. It was recorded as MAXCNN. On this basis, all pooling layers of the Alexnet were replaced with average pooling and stochastic pooling respectively to build another 2 models, which were recorded as MEACNN and STOCNN respectively. The structures of four models were shown in Table 2. The experiment mainly included the following three aspects: (1) the identification results on the training and validation set; (2) the identification results on the test set; (3) identification performance of various models in each rock category.

Table 2. Model structure configuration.

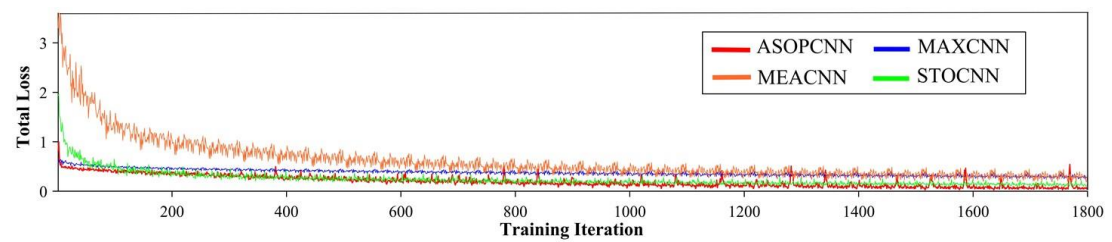
ASOPCNN	MAXCNN	MEACNN	STOCNN
Convolution 11×11 -96 filters	Convolution 11×11 -96 filters	Convolution 11×11 -96 filters	Convolution 11×11 -96 filters
Adaptive pooling	Max pooling	Mean pooling	Stochastic pooling
2 Group convolution	2 Group convolution	2 Group convolution	2 Group convolution
5×5 -128 filters	5×5 -128 filters	5×5 -128 filters	5×5 -128 filters
Adaptive pooling	Max pooling	Mean pooling	Stochastic pooling
Convolution 3×3 -384 filters	Convolution 3×3 -384 filters	Convolution 3×3 -384 filters	Convolution 3×3 -384 filters
2 Group convolution	2 Group convolution	2 Group convolution	2 Group convolution
3×3 -192 filters	3×3 -192 filters	3×3 -192 filters	3×3 -192 filters
2 Group convolution	2 Group convolution	2 Group convolution	2 Group convolution
3×3 -128 filters	3×3 -128 filters	3×3 -128 filters	3×3 -128 filters
Second order pooling	Max pooling	Mean pooling	Stochastic pooling
3 Fully Connected 4096 4096 6	3 Fully Connected 4096 4096 6	3 Fully Connected 4096 4096 6	3 Fully Connected 4096 4096 6
Softmax	Softmax	Softmax	Softmax

In order to speed up the training process, the initial parameters were set by means of transfer learning. The pre-training parameters in the original Alexnet were applied to the convolution layers of the four models. The training set contained 4212 pictures, and the validation set contained 1200 pictures. Mini-batch method was used in the training process, and mini-batch gradient descent method was used to update parameters. 70 pictures were trained in each batch, with 60 iterations in each round. one verification operation was performed every 10 iterations. The learning rate was set to 5×10^{-4} . The feature selection number of second-order pooling layer was set to 9216. A total of 30 cycles, 1800 iterations, were conducted in the whole process.

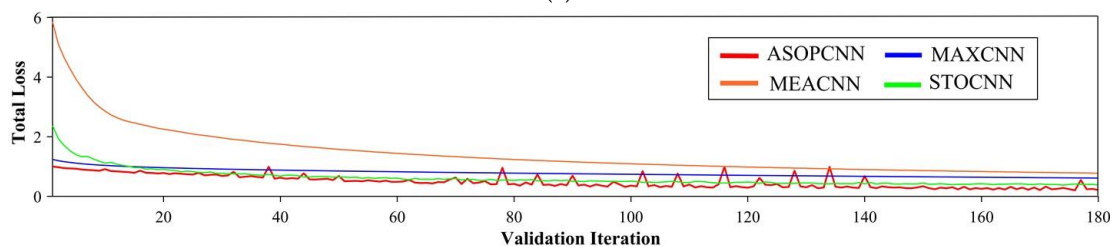
4.1. Training and Validation Results

The total loss is a non-negative function used to measure the difference between predicted values and real values. The smaller the total loss is, the better the training effect and robustness are. Figure 11 showed the loss decline of four models in the training process. In order to compare different decline processes conveniently, the initial loss of ASOPCNN was taken as the benchmark, and the loss values of the other models were divided by the benchmark value for unified processing. It can be seen from Figure 11a that the total loss of ASOPCNN on the training and verification sets was the smallest, indicating that the training effect of ASOPCNN was the best. It can be seen from Figure 11b that the difference between ASOPCNN and the other three models was that the loss fluctuation was more obvious. This was because the pooling layers of ASOPCNN contained random initial parameters. The quality of these parameters had a greater impact on the training performance. Figure 12 showed the changing process of model accuracy. It can be seen that the change rate of

ASOPCNN was slower than that of STOCNN at the initial stage. With the increase in the iteration number, the rate of ASOPCNN exceeded that of the other three models and became the largest. In order to quantitatively compare the final convergence accuracy, the mean accuracy of the last round was taken as the final accuracy, and the results were shown in Table 3. The ASOPCNN model had the highest accuracy on both training and verification sets, which were 0.9286 and 0.8671, respectively. It showed that adaptive and second-order pooling methods were helpful for optimization of training process.

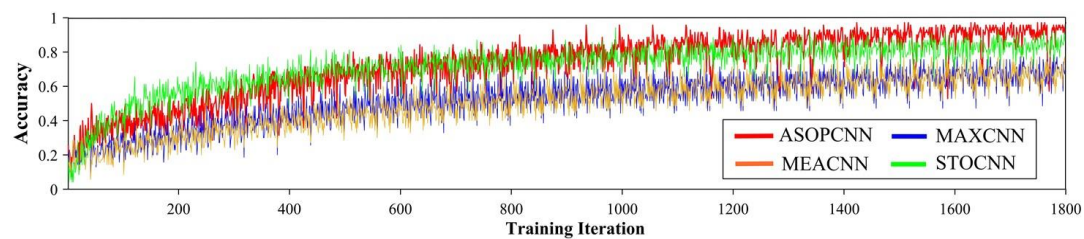


(a)

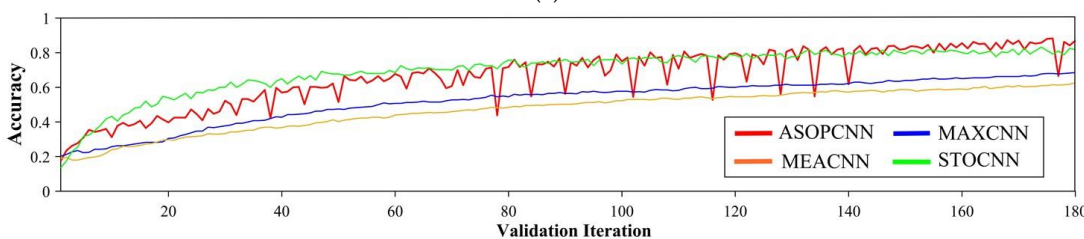


(b)

Figure 11. (a) The loss decline process of four models on the training set; (b) the loss decline process of four models on the verification set.



(a)



(b)

Figure 12. (a) The changing process of model accuracy on the training set; (b) the changing process of model accuracy on the verification set.

Table 3. The final accuracy on training and verification sets.

Model	ASOPCNN	MAXCNN	MEACNN	STOCNN
Training set	0.9286	0.7086	0.6657	0.8412
Validation set	0.8671	0.6808	0.6163	0.8075

4.2. Testing Results

The performance of models on unknown test sets can represent their actual application effects. There were 586 images of six rock types in the rock image test set. Figure 13 showed the accuracy of the four models on the test set. The accuracy of ASOPCNN was 0.8908, higher than MAXCNN (0.6911), MEACNN (0.6297) and STOCNN (0.7696), indicating that the adaptive and second-order pooling methods were more suitable for CNN than traditional pooling methods. In order to visualize identification results, we randomly selected an image from each rock type in the test set for display, as shown in Figure 14. The identification probability corresponding to each image was shown in Table 4. It can be seen that the identification accuracy of each type of rock images was higher than 0.8, which indicated that the ASOPCNN had high identification confidence coefficients and can be used as a reliable model for rock thin section image identification.

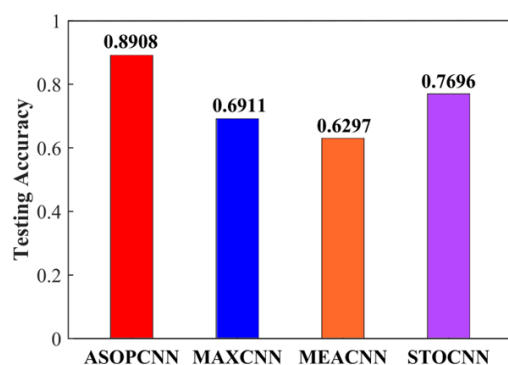


Figure 13. Accuracy of each model on the test set.

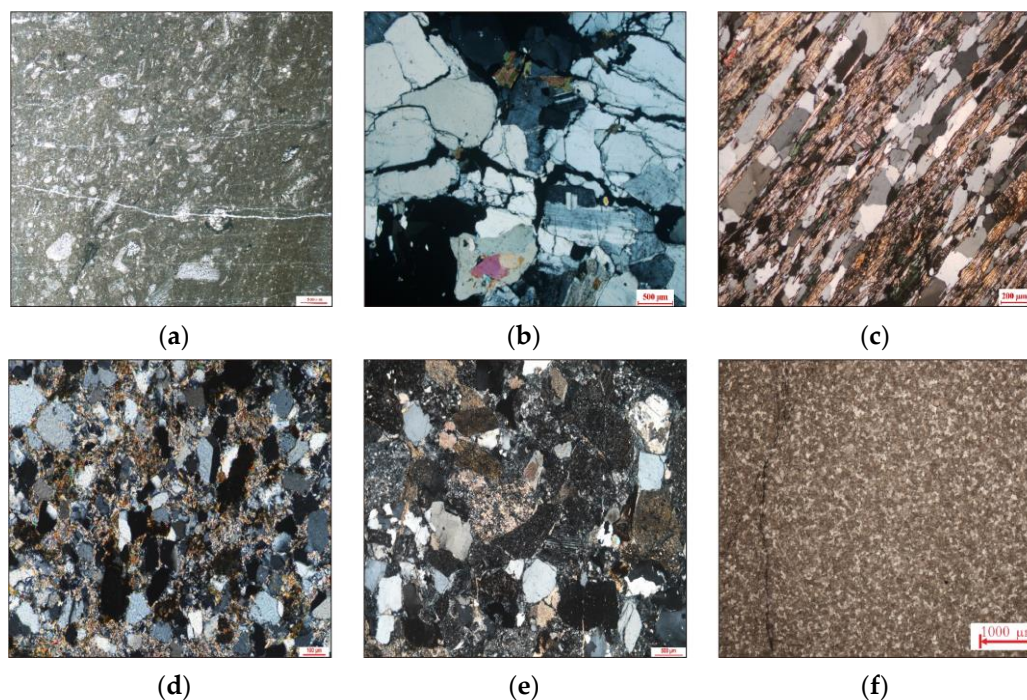


Figure 14. Prediction results of partial images on the test set; (a) wackestone; (b) granite; (c) schist; (d) quartz sandstone; (e) conglomerate; (f) crystalline dolomite.

Confusion matrixes can show the identification performance of each specific category. The confusion matrix obtained by the four models was shown in Figure 15. The elements on the diagonal line were correct identification numbers of each rock type. The correct number of rock images identified by ASOPCNN for each type were 99, 81, 76, 83, 90 and 93, which

were higher than the other three models. From color depths of elements in the confusion matrix, it can be seen that the probability of misclassification between conglomerate and quartz sandstone was greater. From Figure 14d,e, conglomerate and quartz sandstone were very similar. Their main mineral compositions were quartz and feldspar, and their structures were clastic structures. The main difference was the mineral particle size. The numbers of misclassified samples in each category of the ASOPCNN were less than that of the other three models, indicating that the ASOPCNN can distinguish some image samples that were difficult for the others to a certain extent.

Table 4. Identification probability of rock images in Figure 14, the meaning of digits: 1- wackestone, 2-granite; 3-schist, 4-quartz sandstone, 5-conglomerate, 6-crystalline dolomite.

Rock Category	Wackestone	Granite	Schist	Quartz Sandstone	Conglomerate	Crystalline Dolomite
1	0.9241	0.0156	0.0326	0.0123	0.0062	0.0092
2	0.0658	0.8606	0.0369	0.0223	0.0056	0.0088
3	0.0033	0.0015	0.9341	0.0061	0.0089	0.0461
4	0.0031	0.0026	0.0011	0.9073	0.0030	0.0829
5	0.0023	0.1356	0.0011	0.0010	0.8483	0.0117
6	0.0033	0.0011	0.0010	0.0010	0.0918	0.9028

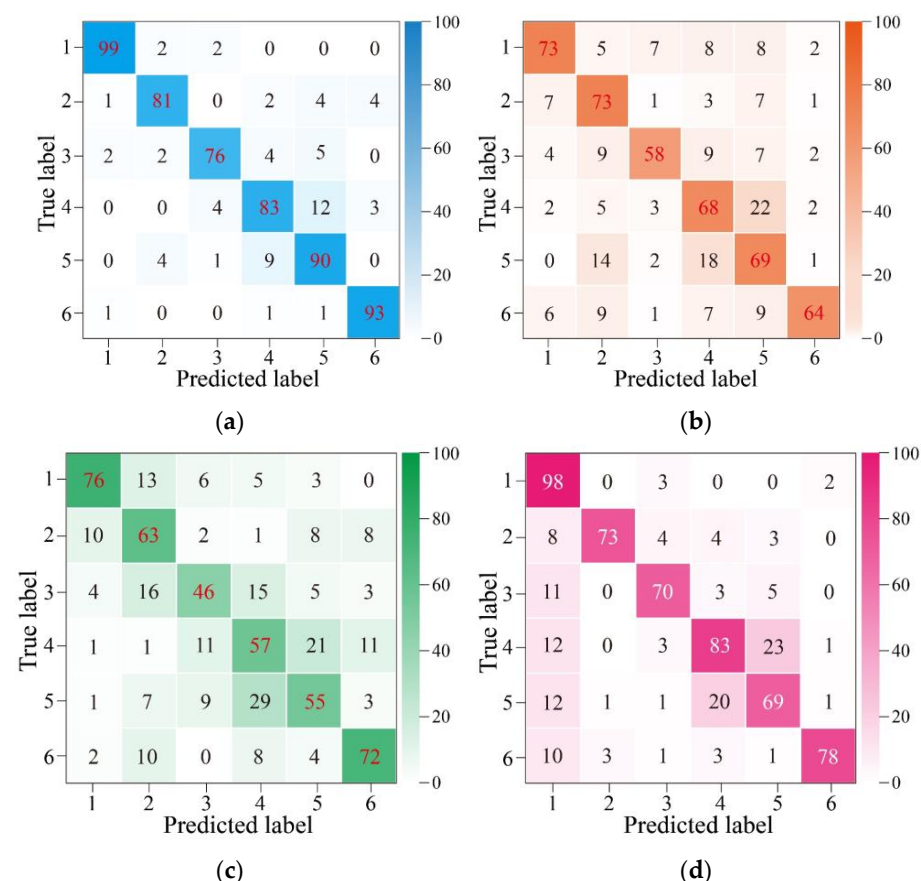


Figure 15. Confusion matrixes of each identification model on the test set, the meaning of digits: 1-wackestone, 2-granite, 3-schist, 4-quartz sandstone, 5-conglomerate, 6-crystalline dolomite. (a) ASOPCNN; (b) MAXCNN; (c) MEACNN; (d) STOCNN.

Figure 16 showed the comparison of the precision, recall and f1 score on the test set. It can be seen that the evaluation indicators of all models showed a similar trend of first falling and then rising, which indicated that the wackestone and crystalline dolomite were easier to identify than the other four types. In Figure 16a, ASOPCNN showed better

performance in the recall rate. From Figure 16b, we can see that the precision rates of ASOPCNN in wackstone, schist, sandstone and conglomerate were higher. Figure 15c showed that ASOPCCNN was higher in F1 score. From the above results, it can be seen that ASOPCCNN had superior classification performance on the whole. Figure 17 showed the ROC curve of the four models on six rock categories. It can be seen that the AUC value of the ASOPCNN was the highest in the categories of wackestone, schist, quartz sandstone, conglomerate and crystalline dolomite. It indicated that the ASOPCNN had the better performance. In Figure 17b, although the AUC value of the ASOPCNN was lower than that of the STOCNN for the granite category, the correct identification number was greater in Figure 15a,d. It indicated that the identification probability values of ASOPCNN were slightly lower than that of STOCNN.

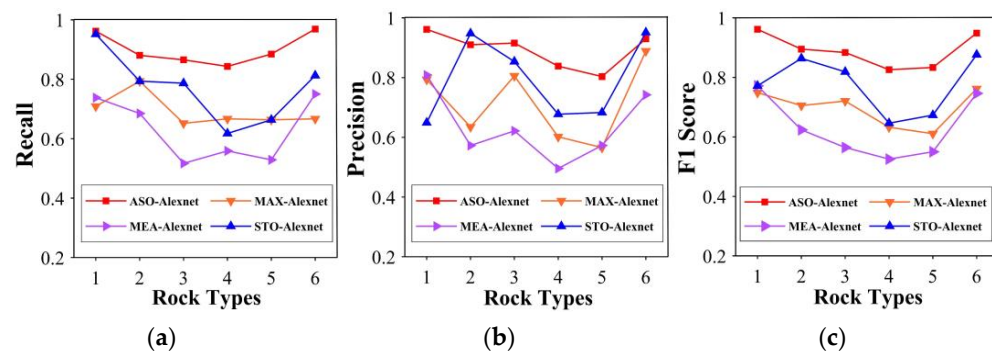


Figure 16. Performance of various rock types evaluated by different metrics, the meaning of digits: 1-wackestone, 2-granite, 3-schist, 4-quartz sandstone, 5-conglomerate, 6-crystalline dolomite. (a) Recall rate; (b) precision rate; (c) F1 score.

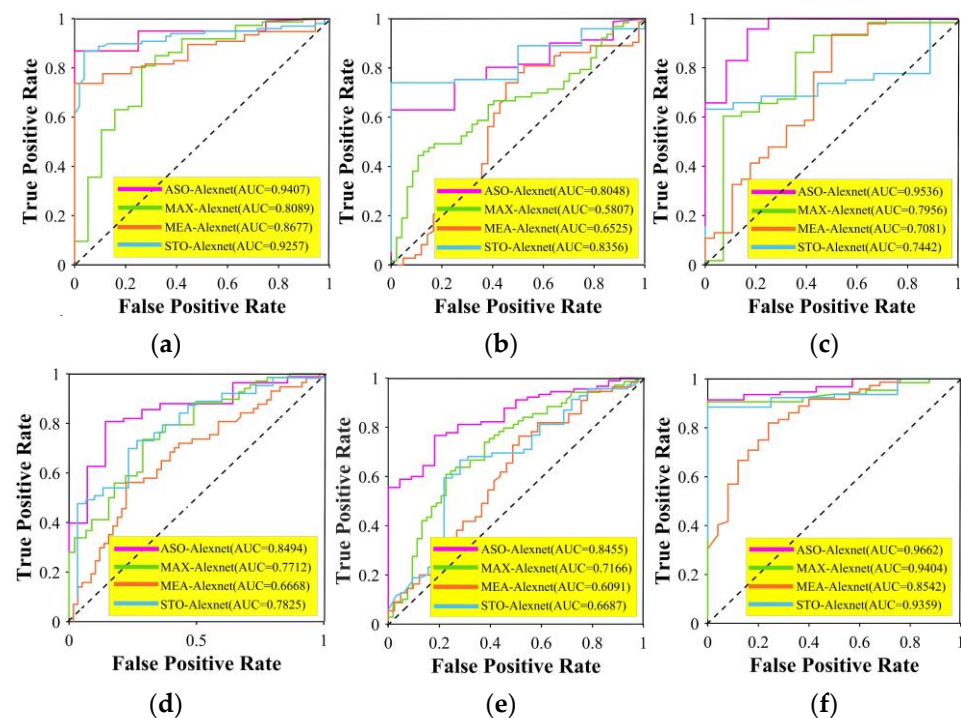


Figure 17. The ROC and AUC values of the four models for six rock categories. (a) wackestone; (b) granite; (c) scholar; (d) quartz sandstone; (e) conglomerate; (f) crystalline dolomite.

4.3. Discussion and Analysis

According to the above results, the pooling layers of CNN can be divided into sampling pooling layers and classification pooling layers, and the adaptive pooling and second-order

pooling methods can be used respectively. It can provide better performance than using traditional pooling methods. This was the explain of experimental results. In fact, from the theoretical analysis, the performance of adaptive and second-order pooling methods was generally not worse than traditional pooling methods. This is because the adaptive pooling method can be transformed into the traditional pooling methods through training, and the second-order pooling method completely contains all the information of the traditional pooling methods.

Adaptive pooling can be seen as a collection of multiple pooling methods. In the adaptive pooling method, a pooling kernel was configured for each feature map. The calculation process is expressed as Equation (8). The parameters in the pooling kernel can be adjusted according to propagation errors. When the parameters are equal, the Equation (8) can transform into the Equation (24), representing the average pooling.

$$O = \frac{\sum_{i=1}^m \sum_{j=1}^n I_{ij}}{m * n} \quad (24)$$

When there was only one non-zero value in pooling kernels, if the non-zero value corresponded to the maximum value, then the Equation (8) can transform into the Equation (25), representing the max pooling.

$$O = \max\{I_{ij}\}, (i = 1, 2, \dots, m; j = 1, 2, \dots, n) \quad (25)$$

When the non-zero value did not correspond to the maximum value, the Equation (8) can transform into the Equation (26), representing the stochastic pooling.

$$O = \text{rand}\{I_{ij}\}, (i = 1, 2, \dots, m; j = 1, 2, \dots, n) \quad (26)$$

When the parameter had more than one non-zero value, the adaptive pooling can also be regarded as other non-special pooling methods. Therefore, the adaptive pooling method can be transformed into the traditional pooling methods through training.

The purpose of classification pooling layer is to integrate feature map information to form a global vector. The integration process of the traditional pooling methods was shown in Figure 8. It only counted first-order feature information. Each feature map was independent and uncorrelated. The integration process of the second-order pooling methods was shown in Figure 18. Second-order pooling was to add more feature information on the basis of guaranteeing the original first-order feature information. Outer products between feature pairs were calculated to obtain second-order correlation. As shown in the yellow diagonal part of the second-order feature block, the diagonal elements were the square of the original first-order feature elements, the feature information carried by them remained unchanged. It can be seen that the first-order feature information extracted by the traditional methods took a small proportion of the information extracted by the second-order pooling method. The upper triangle part of the second-order feature block carried the second-order feature information of feature pairs, which had better feature representation ability than first-order feature information [50]. Therefore, the second-order pooling method completely contained all the information of the traditional pooling methods.

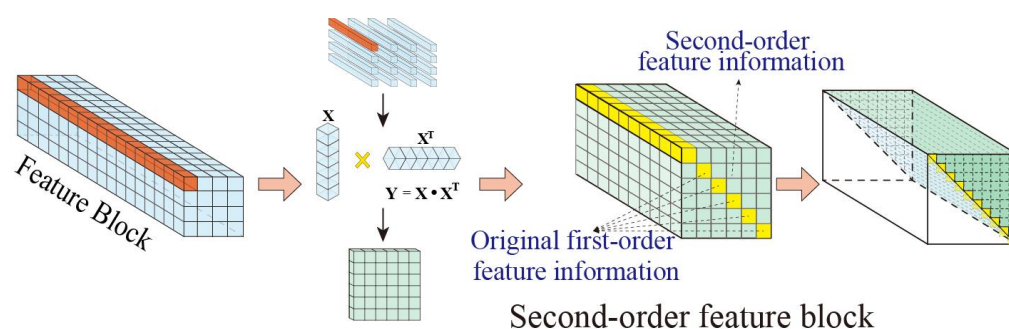


Figure 18. Second-order pooling contained the feature information of traditional pooling methods.

5. Conclusions

In this paper, the shortcomings of traditional pooling methods in the process of rock identification were analyzed, and the adaptive pooling and second-order pooling methods were proposed. The theoretical advantages of these two pooling methods applied to the CNN were analyzed, and then the changing process of forward and back propagation of these two pooling methods was deduced. On the basis of a visual CNN framework, Alexnet, a rock identification model called ASOPCNN was constructed by using the two pooling methods. There were 5998 images for model training, including 6 common rock categories. The rock images were collected from the symposium on Microscopic Images of Rocks in the open-source database, China Scientific Data. Three models were set for comparison, which respectively used the max pooling, average pooling and stochastic pooling methods. The accuracy rate, precision rate, recall rate, F1 score, and AUC values were used to evaluate the model performance. In the results, the total loss of the ASOPCNN is minimum when the training process converges, and the accuracy of ASOPCNN model in training and validation sets are 92.86% and 86.71%, respectively. In the test set of 586 pictures, the identification accuracy of ASOPCNN reaches 89.08%. It is at least 12% higher than the models using traditional pooling methods. The precision rate, recall rate, F1 score and AUC values of the ASOPCNN are 89.33%, 89.07%, 0.8914 and 0.8934, respectively. The experimental results show that the performance of the ASOPCNN is better than the other three models using traditional pooling methods. It also shows that the ASOPCNN could be a reliable model in rock identification of rock thin section images. In addition, among the six rock types in this paper, the wackestone and crystalline dolomite are easier to distinguish than granite, schist, conglomerate and quartz sandstone. Conglomerate and quartz sandstone are more likely to be misclassified because they are similar in mineral compositions and structures. The adaptive and second-order pooling methods were generally not worse than the traditional pooling methods. This is because the adaptive pooling method can be transformed into the traditional pooling methods through training, and the second-order pooling method completely contains all the information of the traditional pooling methods.

Author Contributions: Conceptualization, Z.Z.; Methodology, H.Y.; Supervision, Z.Z. and X.C.; Data curation, H.Y.; Formal analysis, H.Y.; Funding acquisition, Z.Z. and X.C.; Validation, Z.Z. and X.C.; Writing—original draft, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Key Research and Development Program of China (Grant No.: 2022YFC2903901) and the National Natural Science Foundation of China (Grant No.: 52274249).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this paper can be obtained from the symposium on Microscopic Images of Rocks in the open-source database, China Scientific Data.

Acknowledgments: We would like to acknowledge the editors and reviewers for their invaluable comments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The specific derivation process of forward and back propagation of adaptive pooling method is as follows. Since each feature map performs independent but identical pooling operations, only a single feature map is calculated. The feature map

$$x = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1a} \\ x_{21} & x_{22} & \cdots & x_{2a} \\ \cdots & \cdots & \ddots & \cdots \\ x_{a1} & x_{a2} & \cdots & x_{aa} \end{bmatrix}, \text{ called the target feature map, is obtained from the convo-}$$

lution layer. The size of the target feature map is a . The adaptive pooling method is used

for down sampling operation. The pooling kernel is $w = \begin{bmatrix} w_{11} & \cdots & w_{1b} \\ \cdots & \ddots & \cdots \\ w_{b1} & \cdots & w_{bb} \end{bmatrix}$, where b is

the size of the pooling kernel. The result feature map obtained after pooling operation is

$$y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & y_{2c} \\ \cdots & \cdots & \ddots & \cdots \\ y_{c1} & y_{c2} & \cdots & y_{cc} \end{bmatrix}, \text{ where } c \text{ is the size of the result feature map. The element of } y$$

can be calculated by Equation (A1).

$$\begin{aligned} y_{11} &= w_{11}x_{11} + w_{12}x_{12} + w_{13}x_{13} + \cdots + w_{33}x_{33} + \cdots + w_{bb}x_{bb} \\ y_{12} &= w_{11}x_{1,1+s} + w_{12}x_{1,2+s} + w_{13}x_{1,3+s} + \cdots + w_{33}x_{3,3+s} + \cdots + w_{bb}x_{b,b+s} \\ y_{21} &= w_{11}x_{1+s,1} + w_{12}x_{1+s,2} + w_{13}x_{1+s,3} + \cdots + w_{33}x_{3+s,3} + \cdots + w_{bb}x_{b+s,b} \\ y_{22} &= w_{11}x_{1+s,1+s} + w_{12}x_{1+s,2+s} + w_{13}x_{1+s,3+s} + \cdots + w_{33}x_{3+s,3+s} + \cdots + w_{bb}x_{b+s,b+s} \\ &\cdots \end{aligned} \quad (\text{A1})$$

The above equations can be equivalently rewritten as a convolution calculation equation as follows.

$$y_{i,j} = \sum_{m=1}^b \sum_{n=1}^b w_{m,n} \cdot x_{s \cdot (i-1) + m, s \cdot (j-1) + n} \quad (\text{A2})$$

where x is regarded as a Convolution operator; $y_{i,j}$ represents the element in the row i and column j of the result feature map. During the back propagation, the partial derivative

matrix of total loss L to elements of pooling kernels is $\delta w_{i,j} = \begin{bmatrix} \delta w_{11} & \cdots & \delta w_{1b} \\ \cdots & \ddots & \cdots \\ \delta w_{b1} & \cdots & \delta w_{bb} \end{bmatrix}$.

$$\text{Variables } \delta y_{i,j} = \begin{bmatrix} \delta y_{11} & \delta y_{12} & \cdots & \delta y_{1c} \\ \delta y_{21} & \delta y_{22} & \cdots & \delta y_{2c} \\ \cdots & \cdots & \ddots & \cdots \\ \delta y_{c1} & \delta y_{c2} & \cdots & \delta y_{cc} \end{bmatrix} \text{ and } \delta x_{i,j} = \begin{bmatrix} \delta x_{11} & \delta x_{12} & \cdots & \delta x_{1a} \\ \delta x_{21} & \delta x_{22} & \cdots & \delta x_{2a} \\ \cdots & \cdots & \ddots & \cdots \\ \delta x_{a1} & \delta x_{a2} & \cdots & \delta x_{aa} \end{bmatrix} \text{ can}$$

be obtained from similar definitions. The Equation (A3) can be obtained from the chain derivation rule.

$$\frac{\partial L}{\partial w_{i,j}} = \frac{\partial L}{\partial y_{i,j}} \cdot \frac{\partial y_{i,j}}{\partial w_{i,j}} \quad (\text{A3})$$

The relationship between $y_{i,j}$ and $w_{i,j}$ is shown in Equation (A1). After the derivation of both sides of Equation (A1), Equation (A4) can be obtained by combining Equation (A3).

$$\begin{aligned}
 \delta w_{11} &= \delta y_{11}x_{11} + \delta y_{12}x_{1,1+s} + \delta y_{21}x_{1+s,1} + \delta y_{22}x_{1+s,1+s} + \cdots \\
 \delta w_{12} &= \delta y_{11}x_{12} + \delta y_{12}x_{1,2+s} + \delta y_{21}x_{1+s,2} + \delta y_{22}x_{1+s,2+s} + \cdots \\
 \delta w_{13} &= \delta y_{11}x_{13} + \delta y_{12}x_{1,3+s} + \delta y_{21}x_{1+s,3} + \delta y_{22}x_{1+s,3+s} + \cdots \\
 &\vdots \\
 \delta w_{33} &= \delta y_{11}x_{33} + \delta y_{12}x_{3,3+s} + \delta y_{21}x_{3+s,3} + \delta y_{22}x_{3+s,3+s} + \cdots \\
 &\vdots \\
 \delta w_{bb} &= \delta y_{11}x_{bb} + \delta y_{12}x_{b,b+s} + \delta y_{21}x_{b+s,b} + \delta y_{22}x_{b+s,b+s} + \cdots
 \end{aligned} \tag{A4}$$

Equation (A4) can be equivalently written as the following matrix form.

$$\begin{bmatrix} \delta w_{11} & \cdots & \delta w_{1b} \\ \vdots & \ddots & \vdots \\ \delta w_{b1} & \cdots & \delta w_{bb} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1a} \\ x_{21} & x_{22} & \cdots & x_{2a} \\ \vdots & \vdots & \ddots & \vdots \\ x_{a1} & x_{a2} & \cdots & x_{aa} \end{bmatrix} \otimes \begin{bmatrix} \delta y_{11} & 0 & \cdots & 0 & \delta y_{12} & \cdots & \delta y_{1c} \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \delta y_{21} & 0 & \cdots & 0 & \delta y_{22} & \cdots & \delta y_{2c} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \delta y_{c1} & 0 & \cdots & 0 & \delta y_{c2} & \cdots & \delta y_{cc} \end{bmatrix} \tag{A5}$$

The upper equation is recorder as $\delta w_{i,j} = \delta y_{i,j} \otimes A$, where \otimes is the convolution operation symbol. Matrix A is a convolution operator. It is formed by inserting zero vectors with the same dimension between each row and column of the matrix. The number of inserting zero vectors was $s-1$. Through induction and summary, Equation (A5) can be equivalent to the following equation.

$$\delta w_{i,j} = \sum_{m=1}^c \sum_{n=1}^c x_{s \cdot (m-1) + i, s \cdot (n-1) + j} \cdot \delta y_{m,n} \tag{A6}$$

where $\delta y_{m,n}$ is the convolution operator. Parameters of the pooling kernel can be updated by gradient descent algorithm as follows.

$$w_{i,j} = w_{i,j} - \eta \cdot \delta w_{i,j} \tag{A7}$$

where η is the learning rate. If $w_{i,j} \geq 1$, then $w_{i,j} = 1$; if $w_{i,j} \leq 0$, then $w_{i,j} = 0$. The Equation (A8) can be obtained from the chain derivation rule.

$$\frac{\partial L}{\partial x_{i,j}} = \frac{\partial L}{\partial y_{i,j}} \cdot \frac{\partial y_{i,j}}{\partial x_{i,j}} \tag{A8}$$

To facilitate inductive solution, let $s = 2$. After the derivation of both sides of Equation (A1), Equation (A9) can be obtained by combining Equation (A8).

$$\begin{aligned}
 \delta x_{11} &= \delta y_{11}w_{11} \\
 \delta x_{12} &= \delta y_{11}w_{12} \\
 \delta x_{13} &= \delta y_{11}w_{13} + \delta y_{12}w_{11} \\
 &\vdots \\
 \delta x_{21} &= \delta y_{11}w_{21} \\
 \delta x_{22} &= \delta y_{11}w_{22} \\
 \delta x_{23} &= \delta y_{11}w_{23} + \delta y_{12}w_{21} \\
 &\vdots \\
 \delta x_{31} &= \delta y_{11}w_{31} + \delta y_{21}w_{11} \\
 \delta x_{32} &= \delta y_{11}w_{32} + \delta y_{21}w_{12} \\
 \delta x_{33} &= \delta y_{11}w_{33} + \delta y_{12}w_{31} + \delta y_{21}w_{13} + \delta y_{22}w_{11} \\
 &\dots\dots
 \end{aligned} \tag{A9}$$

Through induction and summary, Equation (A9) can be equivalently rewritten as the following matrix form in the case of $s = 2$.

$$\begin{bmatrix} \delta x_{11} & \delta x_{12} & \cdots & \delta x_{1a} \\ \delta x_{21} & \delta x_{22} & \cdots & \delta x_{2a} \\ \cdots & \cdots & \ddots & \cdots \\ \delta x_{a1} & \delta x_{a2} & \cdots & \delta x_{aa} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & w_{11} & \cdots & w_{1b} & 0 & 0 \\ 0 & 0 & \cdots & \ddots & \cdots & 0 & 0 \\ 0 & 0 & w_{b1} & \cdots & w_{bb} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} \delta y_{cc} & \cdots & \delta y_{c2} & 0 & \delta y_{c1} \\ \cdots & \ddots & \cdots & \cdots & \cdots \\ \delta y_{2c} & \cdots & \delta y_{22} & 0 & \delta y_{21} \\ 0 & \cdots & 0 & 0 & 0 \\ \delta y_{1c} & \cdots & \delta y_{12} & 0 & \delta y_{11} \end{bmatrix} \quad (\text{A10})$$

The upper equation is recorded as $\delta x_{i,j} = B \otimes C$. Matrix B is composed of matrix W with an appropriate number of 0 in the outer circle. Matrix C is a convolution operator. It was formed by rotating the matrix elements by 180 degrees and then inserting an appropriate number of zero vectors between every two columns as same as every two rows. The numbers of zero filling in the outer of matrix B and zero vectors inserted in matrix C are related to the pooling step s and the size c of the result feature map. The calculation formula is as follows.

$$\begin{aligned} \alpha &= s(c-1) \\ \beta &= s-1 \end{aligned} \quad (\text{A11})$$

Through induction and summary, Equation (A10) can be equivalent to the Equation (A12).

$$\delta x_{i,j} = \sum_{m=1}^c \sum_{n=1}^c w_{i-s \cdot m+s, j-s \cdot n+s} \cdot \delta y_{m,n} \cdot f[i-s \cdot (m-1)] \cdot f[j-s \cdot (n-1)] \quad (\text{A12})$$

where $f(x)$ is the judgement function, which equals to 1 while $n < x < b$ and equals to 0 for else.

Appendix B

The specific derivation process of forward and back propagation of second-order pooling method is as follows. The l th original feature map calculated from the last convolution

$$\text{is } x_l = \begin{bmatrix} x_{1,1,l} & x_{1,2,l} & \cdots & x_{1,a,l} \\ x_{2,1,l} & x_{2,2,l} & \cdots & x_{2,a,l} \\ \cdots & \cdots & \ddots & \cdots \\ x_{a,1,l} & x_{a,2,l} & \cdots & x_{a,a,l} \end{bmatrix}, (l = 1, 2, 3, \dots, s_1), \text{ where } a \text{ is the size of the original}$$

feature maps; s_1 is the number of feature maps contained. The original feature maps are split from left to right and top to bottom. Then the elements at the same position in all the original feature maps are integrated to form the local first-order feature vector, which is recorded as $A = (x_{i,j,1}, x_{i,j,2}, x_{i,j,3}, \dots, x_{i,j,s_1}), (i, j = 1, 2, \dots, a)$. The outer products of all local first-order feature vectors are calculated according to the following equation.

$$\begin{bmatrix} x_{i,j,1}^2 & x_{i,j,1}x_{i,j,2} & x_{i,j,1}x_{i,j,3} & \cdots & x_{i,j,1}x_{i,j,s_1} \\ x_{i,j,2}x_{i,j,1} & x_{i,j,2}^2 & x_{i,j,2}x_{i,j,3} & \cdots & x_{i,j,2}x_{i,j,s_1} \\ x_{i,j,3}x_{i,j,1} & x_{i,j,3}x_{i,j,2} & x_{i,j,3}^2 & \cdots & x_{i,j,3}x_{i,j,s_1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{i,j,s_1}x_{i,j,1} & x_{i,j,s_1}x_{i,j,2} & x_{i,j,s_1}x_{i,j,3} & \cdots & x_{i,j,s_1}^2 \end{bmatrix} = (x_{i,j,1}, \dots, x_{i,j,s_1})^T \cdot (x_{i,j,1}, \dots, x_{i,j,s_1}), (i, j = 1, 2, \dots, a) \quad (\text{A13})$$

The above equation is recorded as $y = A^T A$. y is the outer product of the local first-order feature vector, called the second-order feature matrix. The k th second-order fea-

ture matrix can also be recorded as $y_k = \begin{bmatrix} y_{1,1,k} & y_{1,2,k} & y_{1,3,k} & \cdots & y_{1,b,k} \\ y_{2,1,k} & y_{2,2,k} & y_{2,3,k} & \cdots & y_{2,b,k} \\ y_{3,1,k} & y_{3,2,k} & y_{3,3,k} & \cdots & y_{3,b,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{b,1,k} & y_{b,2,k} & y_{b,3,k} & \cdots & y_{b,b,k} \end{bmatrix},$

($k = 1, 2, 3, \dots, s_2$), where b is the size of the second-order feature matrix; s_2 is the number of feature maps contained. Combining Equation (A13), the relationship between the second-order feature matrixes and the original feature maps can be constructed as follows.

$$y_{i,j,k} = \sum_{m=1}^a \sum_{n=1}^a x_{m,n,i} \cdot x_{m,n,j} \cdot I[a(m-1) + n = k], (i, j = 1, 2, \dots, b; k = 1, 2, \dots, s_2) \quad (\text{A14})$$

where $I(\text{condition})$ is the judgment function. If the condition is met, then $I = 1$; otherwise $I = 0$. It is not difficult to see the following variable relations.

$$\begin{aligned} b &= s_1 \\ s_2 &= a^2 \end{aligned} \quad (\text{A15})$$

The second-order feature matrixes are split from left to right and top to bottom. Then the elements at the same position in all second-order feature matrices are integrated to form the second-order feature vectors, which can be recorded as $B = (y_{i,j,1}, y_{i,j,2}, y_{i,j,3}, \dots, y_{i,j,s_2})$ ($i, j = 1, 2, 3, \dots, b$). According to Equation (A15), all second-order feature vectors can be rewritten into matrix form as follows. Because the second-order feature matrixes are symmetric, only the upper triangular part is converted into the corresponding second-order feature vectors.

$$(y_{i,j,1}, y_{i,j,2}, y_{i,j,3}, \dots, y_{i,j,s_2}) \rightarrow \begin{bmatrix} y_{i,j,1} & y_{i,j,2} & y_{i,j,3} & \cdots & y_{i,j,a} \\ y_{i,j,a+1} & y_{i,j,a+2} & y_{i,j,a+3} & \cdots & y_{i,j,2a} \\ y_{i,j,2a+1} & y_{i,j,2a+2} & y_{i,j,2a+3} & \cdots & y_{i,j,3a} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{i,j,a(a-1)+1} & y_{i,j,a(a-1)+2} & y_{i,j,a(a-1)+3} & \cdots & y_{i,j,a^2} \end{bmatrix} \quad (\text{A16})$$

The matrix transformed from the second-order feature vector is called the second-order feature map, and the γ th second-order feature maps can be recorded as

$$z_\gamma = \begin{bmatrix} z_{1,1,\gamma} & z_{1,2,\gamma} & z_{1,3,\gamma} & \cdots & z_{1,c,\gamma} \\ z_{2,1,\gamma} & z_{2,2,\gamma} & z_{2,3,\gamma} & \cdots & z_{2,c,\gamma} \\ z_{3,1,\gamma} & z_{3,2,\gamma} & z_{3,3,\gamma} & \cdots & z_{3,c,\gamma} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{c,1,\gamma} & z_{c,2,\gamma} & z_{c,3,\gamma} & \cdots & z_{c,c,\gamma} \end{bmatrix}, (\gamma = 1, 2, 3, \dots, s_3), \text{ where } c \text{ is the}$$

size of the second-order feature maps; s_3 is the number of feature maps contained. The relationship between the second-order feature maps and the second-order feature matrixes can be constructed by combining Equation (A16).

$$z_{\alpha,\beta,\gamma} = \sum_{k=1}^{s_2} \sum_{i=1}^b \sum_{j=1}^b y_{i,j,k} \cdot I[b(i-1) + j - \frac{i^2-i}{2} = \gamma] \cdot I[c(\alpha-1) + \beta = \gamma], (\alpha, \beta = 1, 2, \dots, c; \gamma = 1, 2, \dots, s_3) \quad (\text{A17})$$

It is not difficult to see the following variable relations.

$$\begin{aligned} c &= a \\ s_3 &= \frac{b^2+b}{2} \end{aligned} \quad (\text{A18})$$

The second order feature matrixes are down sampling to obtain the global feature vector. It can be recorded as $Z = (Z_1, Z_2, Z_3, \dots, Z_{s_4})$, where s_4 is the number of vectors contained. The calculation equation is as follows.

$$Z_p = \frac{\sum_{\alpha=1}^c \sum_{\beta=1}^c z_{\alpha,\beta,p}}{\alpha \cdot \beta}, (p = 1, 2, 3, \dots, s_4) \quad (\text{A19})$$

In the back propagation process, since the entire pooling process doesn't contain parameters, only the derivative of the total loss to each feature map is required. The derivative of the total loss to the global feature vector is $\delta Z = (\delta Z_1, \delta Z_2, \delta Z_3, \dots, \delta Z_{s_4})$, which is obtained through the fully connected layer. The error of the global feature vector is back propagated to the second-order feature matrixes, and the relationship between them is shown in Equation (A19). Then the error sensitivity in the second-order feature maps,

$$\text{recorded as } \delta z_{\gamma} = \begin{bmatrix} \delta z_{1,1,\gamma} & \delta z_{1,2,\gamma} & \delta z_{1,3,\gamma} & \cdots & \delta z_{1,c,\gamma} \\ \delta z_{2,1,\gamma} & \delta z_{2,2,\gamma} & \delta z_{2,3,\gamma} & \cdots & \delta z_{2,c,\gamma} \\ \delta z_{3,1,\gamma} & \delta z_{3,2,\gamma} & \delta z_{3,3,\gamma} & \cdots & \delta z_{3,c,\gamma} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta z_{c,1,\gamma} & \delta z_{c,2,\gamma} & \delta z_{c,3,\gamma} & \cdots & \delta z_{c,c,\gamma} \end{bmatrix}, (\gamma = 1, 2, 3, \dots, s_3),$$

can be calculated as follows.

$$\delta z_{\alpha,\beta,\gamma} = \frac{\delta z_{\gamma}}{\alpha \cdot \beta}, (\alpha, \beta = 1, 2, 3, \dots, c; \gamma = 1, 2, 3, \dots, s_3) \quad (\text{A20})$$

It can be seen from Equation (A16) that the second-order feature maps need to convert into vectors when the error is transmitted from them to the second-order feature vectors. Therefore, the second-order feature maps are converted into the second-order feature vectors in the following way.

$$\begin{bmatrix} \delta z_{1,1,\gamma} & \delta z_{1,2,\gamma} & \delta z_{1,3,\gamma} & \cdots & \delta z_{1,c,\gamma} \\ \delta z_{2,1,\gamma} & \delta z_{2,2,\gamma} & \delta z_{2,3,\gamma} & \cdots & \delta z_{2,c,\gamma} \\ \delta z_{3,1,\gamma} & \delta z_{3,2,\gamma} & \delta z_{3,3,\gamma} & \cdots & \delta z_{3,c,\gamma} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta z_{c,1,\gamma} & \delta z_{c,2,\gamma} & \delta z_{c,3,\gamma} & \cdots & \delta z_{c,c,\gamma} \end{bmatrix} \rightarrow (\delta z_{1,1,\gamma}, \delta z_{1,2,\gamma}, \dots, \delta z_{c,c,\gamma}), (\gamma = 1, 2, 3, \dots, s_3) \quad (\text{A21})$$

All the second-order feature vectors are combined into the second-order feature matrixes in order, and the error remains unchanged. Because the second-order feature matrixes are symmetric and the elements of the lower triangular part of the matrixes are not involved in the calculation, the error of the elements of the lower triangular part equals to 0. The equation for calculating the error sensitivity of the second-order feature matrixes can be deduced as follows by replacing δz with δy in Equation (A22).

$$\delta y_{i,j,k} = \begin{cases} \sum_{\alpha=1}^c \sum_{\beta=1}^c \delta z_{\alpha,\beta,b(i-1)+j-(i^2-i)/2} \cdot I[b(\alpha-1) + \beta = k], i < j \\ 0, i > j \end{cases} \quad (\text{A22})$$

The second-order feature matrixes back propagate the error to the local first-order feature vectors. The relationship between them is shown in Equation (A13). The error sensitivity of the local first-order feature vectors can be obtained by derivation of Equation (A13). In the process of transforming the local first-order feature vectors into the original feature maps, the element value does not change but the position changes. Finally, the relationship between the error sensitivity of the second-order feature matrixes and original feature maps can be established as follows.

$$\delta x_{m,n,l} = \sum_{i=1}^b \sum_{j=1}^b \delta y_{i,j,s_2 \cdot (m-1) + n} \cdot x_{m,n,j} \cdot I(i=l) \cdot [I(i=m) \cdot I(j=n) + 1], (m, n = 1, 2, \dots, a; l = 1, 2, \dots, s_1) \quad (\text{A23})$$

References

- Xu, Z.; Ma, W.; Lin, P.; Hua, Y. Deep learning of rock microscopic images for intelligent lithology identification: Neural network comparison and selection. *J. Rock. Mech. Geotech. Eng.* **2022**, *14*, 1140–1152. [\[CrossRef\]](#)
- Liu, N.; Huang, T.; Gao, J.; Xu, Z.; Wang, D.; Li, F. Quantum-Enhanced Deep Learning-Based Lithology Interpretation from Well Logs. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4503213. [\[CrossRef\]](#)
- Pi, Z.; Zhou, Z.; Li, X.; Wang, S. Digital image processing method for characterization of fractures, fragments, and particles of soil/rock-like materials. *Mathematics* **2021**, *9*, 815. [\[CrossRef\]](#)
- Zhou, Z.; Lu, J.; Cai, X.; Rui, Y.; Tan, L. Water saturation effects on mechanical performances and failure characteristics of rock-concrete disc with different interface dip angles. *Constr. Build. Mater.* **2022**, *324*, 126684. [\[CrossRef\]](#)
- Yin, J.; Lu, J.; Tian, F.; Wang, S. Pollutant Migration Pattern during Open-Pit Rock Blasting Based on Digital Image Analysis Technology. *Mathematics* **2022**, *10*, 3205. [\[CrossRef\]](#)
- Zhou, Z.; Zang, H.; Cao, W.; Du, X.; Chen, L.; Ke, C. Risk assessment for the cascading failure of underground pillar sections considering interaction between pillars. *Int. J. Rock. Mech. Min. Sci.* **2019**, *124*, 104142. [\[CrossRef\]](#)
- Xue, Y.; Li, X.; Li, G.; Qiu, D.; Gong, H.; Kong, F. An analytical model for assessing soft rock tunnel collapse risk and its engineering application. *Geomech. Eng.* **2020**, *23*, 441–454.
- Li, D.; Zhao, J.; Ma, J. Experimental Studies on Rock Thin-Section Image Classification by Deep Learning-Based Approaches. *Mathematics* **2022**, *10*, 2317. [\[CrossRef\]](#)
- Zhang, B. Application of thin section micro-image in identification of rock. *Petrochemical. Ind. Technol.* **2016**, *23*, 108.
- Xu, C.; Li, L.; Zhang, J.; He, L.; Zhang, G.; Wang, Y. Application of X-ray Fluorescence Spectrometry and Electron Microprobe in the Identification of Intermediate-Felsic Volcanic Rocks. *Rock. Miner. Anal.* **2016**, *35*, 626–633.
- Singh, N.; Singh, T.; Tiwary, A.; Sarkar, K. Textural identification of basaltic rock mass using image processing and neural network. *Comput. Geosci.* **2010**, *14*, 301–310. [\[CrossRef\]](#)
- Mlynarczyk, M.; Gorszczyk, A.; Slipek, B. The application of pattern recognition in the automatic classification of microscopic rock images. *Comput. Geosci.* **2013**, *60*, 126–133. [\[CrossRef\]](#)
- Chatterjee, S. Vision-based rock-type classification of limestone using multi-class support vector machine. *Appl. Intell.* **2013**, *39*, 14–27. [\[CrossRef\]](#)
- Patel, A.; Chatterjee, S. Computer vision-based limestone rock-type classification using probabilistic neural network. *Geosci. Front.* **2016**, *7*, 53–60. [\[CrossRef\]](#)
- Zhang, Y.; Li, M.; Han, S.; Ren, Q.; Shi, J. Intelligent Identification for Rock-Mineral Microscopic Images Using Ensemble Machine Learning Algorithms. *Sensors* **2019**, *19*, 3914. [\[CrossRef\]](#) [\[PubMed\]](#)
- Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [\[CrossRef\]](#)
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
- Esteva, A.; Kuprel, B.; Novoa, R. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [\[CrossRef\]](#)
- Pan, S.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [\[CrossRef\]](#)
- Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [\[CrossRef\]](#)
- Bamford, T.; Esmaeili, K.; Schoellig, A. A deep learning approach for rock fragmentation analysis. *Int. J. Rock. Mech. Min. Sci.* **2021**, *145*, 104839. [\[CrossRef\]](#)
- Wang, P.; Wang, S.; Zhu, C.; Zhang, Z. Classification and extent determination of rock slope using deep learning. *Geomech. Geophys. Geo.* **2020**, *6*, 33. [\[CrossRef\]](#)
- Li, H.; Hu, Q.; Mao, Y.; Niu, F.; Liu, C. Deep Learning-based Model for Automatic Salt Rock Segmentation. *Rock. Mech. Rock. Eng.* **2022**, *55*, 3735–3747. [\[CrossRef\]](#)
- Zhao, J.; Wang, F.; Cai, J. 3D tight sandstone digital rock reconstruction with deep learning. *J. Pet. Sci. Eng.* **2021**, *207*, 109020. [\[CrossRef\]](#)
- Cao, D.; Ji, S.; Cui, R.; Liu, Q. Multi-task learning for digital rock segmentation and characteristic parameters computation. *J. Pet. Sci. Eng.* **2022**, *208*, 109202. [\[CrossRef\]](#)
- Li, D.; Zhao, J.; Liu, Z. A Novel Method of Multitype Hybrid Rock Lithology Classification Based on Convolutional Neural Networks. *Sensors* **2022**, *22*, 1574. [\[CrossRef\]](#)
- Polat, O.; Polat, A.; Ekici, T. Automatic classification of volcanic rocks from thin section images using transfer learning networks. *Neural Comput. Appl.* **2021**, *33*, 11531–11540. [\[CrossRef\]](#)
- Alzubaidi, F.; Mostaghimi, P.; Swietojanski, P.; Clark, S.; Armstrong, R. Automated lithology classification from drill core images using convolutional neural networks. *J. Pet. Sci. Eng.* **2021**, *197*, 107933. [\[CrossRef\]](#)

29. Liu, Y.; Zhang, Z.; Liu, X.; Wang, L.; Xia, X. Deep Learning Based Mineral Image Classification Combined with Visual Attention Mechanism. *IEEE Access* **2021**, *9*, 98091–98109. [CrossRef]
30. Ma, H.; Han, G.; Peng, L.; Zhu, L.; Shu, J. Rock thin sections identification based on improved squeeze-and-Excitation Networks model. *Comput. Geosci.* **2021**, *152*, 104780. [CrossRef]
31. dos Anjos, C.; Avila, M.; Vasconcelos, A.; Neta, A.; Medeiros, L.; Evsukoff, A.; Surmas, R.; Landau, L. Deep learning for lithological classification of carbonate rock micro-CT images. *Comput. Geosci.* **2021**, *25*, 971–983. [CrossRef]
32. Su, C.; Xu, S.; Zhu, K.; Zhang, X. Rock classification in petrographic thin section images based on concatenated convolutional neural networks. *Earth Sci. Inform.* **2020**, *13*, 1477–1484. [CrossRef]
33. Seo, W.; Kim, Y.; Sim, H.; Song, Y.; Yun, T. Classification of igneous rocks from petrographic thin section images using convolutional neural network. *Earth Sci. Inform.* **2022**, *15*, 1297–1307. [CrossRef]
34. Xu, Z.; Shi, H.; Lin, P.; Liu, T. Integrated lithology identification based on images and elemental data from rocks. *J. Pet. Sci. Eng.* **2021**, *205*, 108853. [CrossRef]
35. Saeedan, F.; Weber, N.; Goesele, M.; Roth, S. Detail-Preserving Pooling in Deep Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
36. Li, P.; Xie, J.; Wang, Q.; Zuo, W. Is Second-order Information Helpful for Large-scale Visual Recognition. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
37. Ma, R.; Liu, C.; Yang, J.; Wang, Y.; Liu, J. A Carbonate Microscopic Image Dataset of the Permo-Carboniferous Taiyuan Formation in the Southern Margin of the North China Block. Science Data Bank. Available online: <https://www.scidb.cn/en/detail?dataSetId=727517165267189760> (accessed on 15 October 2022).
38. Liu, Y.; Hou, M.; Liu, X.; Qi, Z. A Micrograph Dataset of Buried Hills and Overlying Glutenite in Bozhong Sag, Bohai Bay Basin. Science Data Bank. Available online: <https://www.scidb.cn/en/detail?dataSetId=752623639467130880> (accessed on 15 October 2022).
39. Lai, W.; Jiang, J.; Qiu, J.; Yu, J.; Hu, X. Photomicrograph Dataset of Rocks for Petrology Teaching in Nanjing University. Science Data Bank. Available online: <https://www.scidb.cn/en/detail?dataSetId=732953783604084736> (accessed on 15 October 2022).
40. Qi, Z.; Hou, M.; Xu, S.; He, L.; Tang, Z.; Zhang, M. A Carbonate Microscopic Image Dataset of Sinian Dengying Period in Northwestern Margin of Sichuan Basin. Science Data Bank. Available online: <https://www.scidb.cn/en/detail?dataSetId=733012342660399104> (accessed on 15 October 2022).
41. Ma, Q.; Chai, R.; Yang, J.; Du, Y.; Dai, X. A microscopic Image Dataset of Mesozoic Metamorphic Grains Bearing Sandstones from Mid-Yangtze, China. Science Data Bank. Available online: <https://www.scidb.cn/en/detail?dataSetId=727525043063488512> (accessed on 15 October 2022).
42. Cai, W.; Hou, M.; Chen, H.; Liu, Y. A Micrograph Dataset of Terrigenous Clastic Rocks of Upper Devonian Lower Carboniferous Wutong Group in Southern Lower Yangtze. Science Data Bank. Available online: <https://www.scidb.cn/en/detail?dataSetId=732987889075355648> (accessed on 15 October 2022).
43. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
44. Yang, Z.; He, B.; Liu, Y.; Wang, D.; Zhu, G. Classification of rock fragments produced by tunnel boring machine using convolutional neural networks. *Autom. Constr.* **2021**, *125*, 103612. [CrossRef]
45. Yu, Z.; Dai, S.; Xing, Y. Adaptive Saliency Preserving Pooling for Deep Convolutional Neural Networks. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo Workshops, Shanghai, China, 8–12 July 2019.
46. Carreira, J.; Caseiro, R.; Batista, J.; Sminchisescu, C. Free-form region description with second-order pooling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1177–1189. [CrossRef]
47. Zhou, J.; Huang, S.; Wang, M.; Qiu, Y. Performance evaluation of hybrid GA-SVM and GWO-SVM models to predict earthquake-induced liquefaction potential of soil: A multi-dataset investigation. *Eng. Comput.* **2021**, *38*, 4197–4215. [CrossRef]
48. Zhou, J.; Huang, S.; Zhou, T.; Armaghani, D.; Qiu, Y. Employing a genetic algorithm and grey wolf optimizer for optimizing RF models to evaluate soil liquefaction potential. *Artif. Intell. Rev.* **2022**, *55*, 5673–5705. [CrossRef]
49. Zhou, J.; Huang, S.; Qiu, Y. Optimization of random forest through the use of MVO, GWO and MFO in evaluating the stability of underground entry-type excavations. *Tunn. Undergr. Space. Technol.* **2022**, *124*, 104494. [CrossRef]
50. Li, P.; Xie, J.; Wang, Q.; Gao, Z. Towards Faster Training of Global Covariance Pooling Networks by Iterative Matrix Square Root Normalization. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.