

## Research Article

# A Deep Learning Model for Quick and Accurate Rock Recognition with Smartphones

Guangpeng Fan,<sup>1,2</sup> Feixiang Chen ,<sup>1,2</sup> Danyu Chen,<sup>1,2</sup> Yan Li,<sup>1,2</sup> and Yanqi Dong<sup>1,2</sup>

<sup>1</sup>School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China

<sup>2</sup>Engineering Research Center for Forestry-oriented Intelligent Information Processing, National Forestry and Grassland Administration, Beijing 100083, China

Correspondence should be addressed to Feixiang Chen; [bjfxchen@bjfu.edu.cn](mailto:bjfxchen@bjfu.edu.cn)

Received 1 August 2019; Accepted 23 April 2020; Published 19 May 2020

Academic Editor: Zeeshan A. Rana

Copyright © 2020 Guangpeng Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the geological survey, the recognition and classification of rock lithology are an important content. The recognition method based on rock thin section leads to long recognition period and high recognition cost, and the recognition accuracy cannot be guaranteed. Moreover, the above method cannot provide an effective solution in the field. As a communication device with multiple sensors, smartphones are carried by most geological survey workers. In this paper, a smartphone application based on the convolutional neural network is developed. In this application, the phone's camera can be used to take photos of rocks. And the types and lithology of rocks can be quickly and accurately identified in a very short time. This paper proposed a method for quickly and accurately recognizing rock lithology in the field. Based on ShuffleNet, a lightweight convolutional neural network used in deep learning, combined with the transfer learning method, the recognition model of the rock image was established. The trained model was then deployed to the smartphone. A smartphone application for identifying rock lithology was designed and developed to verify its usability and accuracy. The research results showed that the accuracy of the recognition model in this paper was 97.65% on the verification data set of the PC. The accuracy of recognition on the test data set of the smartphone was 95.30%, among which the average recognition time of the single sheet was 786 milliseconds, the maximum value was 1,045 milliseconds, and the minimum value was 452 milliseconds. And the single-image accuracy above 96% accounted for 95% of the test data set. This paper presented a new solution for the rapid and accurate recognition of rock lithology in field geological surveys, which met the needs of geological survey personnel to quickly and accurately identify rock lithology in field operations.

## 1. Introduction

The recognition of rocks is not only an important part of geological survey but also the focus of geological research. The traditional recognition method consists of three steps [1, 2]: firstly, workers collect fresh rock samples in the process of exploration; secondly, after returning to the laboratory, the rock thin section with an area of about  $2 \times 2$  cm is cut from the vertical stratification direction of the rock samples. When one side of the rock samples has been flattened on the grinding machine, it is glued to the carrier glass with glue such as adhesive. Then, the thickness of the other side is smoothed to 0.03 mm, and the cover glass piece is glued with the adhesive. Finally, an image of the rock sheet is viewed under a polarizing microscope by a knowledgeable

or experienced geologist. In this way, the rock type and structural parameters can be determined. This traditional identification method requires the observer to have very rich geological knowledge and experience. In addition, the method has many problems, such as strong subjectivity, long identification period, and poor field identification ability.

With the development of computer vision and image processing technology, great changes have been brought to rock recognition and mineral analysis [3, 4]. Many researchers analyzed the texture, fabric, granularity, and lithology of rocks based on image processing techniques such as image analysis and feature extraction. Patel used the probabilistic neural network (PNN) to develop a lab-scale vision-based model in which color histogram features are used as the input. The model has achieved good recognition

results, and the error of misclassification of limestone is less than 6%. The main limitation of this study is that the classification object is the entire rock sample of multiple rocks, and no further consideration is applied to identify rocks on site [5, 6]. Based on the basalt thin section image, Singh et al. extracted 27 characteristic parameters and identified 300 rock thin sections [7]. The recognition accuracy of the three texture categories is 92.22%, which is improved compared with previous studies, but the classification categories are fewer. Based on the research of image features, Cheng Guojian and Yin Juanjuan applied the support vector machine to realize the image classification of a total of 100 rock thin sections of 4 categories, with an accuracy of 80% [8]. The disadvantage is that the model performed poorly. With the continuous development of deep learning in the field of image intelligent recognition, many researchers used deep learning methods to automatically identify rock images [9–11]. Zhang Ye et al. used the transfer learning method for the first time to automatically identify and classify rock images. They have achieved the effective identification of three types of granite, Chiba, and breccia [12]. However, the experimental data are few and cannot meet the needs of on-site recognition. Li et al. used the transfer learning method to train the sandstone microscopic images to obtain a high-precision sandpaper slice microscopic image classification model [13]. The disadvantage is that the adaptability is poor, and it is only suitable for sandstone recognition. Cheng Guojian, Guo Wenhui and others realized automatic granularity recognition based on the rock thin section image [14]. The accuracy of rock identification is 98.5%. However, the identification objects of this study are rock thin section images, which need to be made in the laboratory and cannot be directly applied to the work site. Based on computer vision and machine learning, Marmo et al. used more than 1,000 carbonate flakes. Based on the gray scale digital image, they set up the multilayer sensory neural network model. Then, network training based on texture data was carried out, and the classification accuracy reached 93.3% [15]. Guo Chao et al. used the original color image of the rock to describe the feature space. Their method was to calculate the standard arithmetic values of different color channels by combining their morphologies [16]. The neural network is used to establish the mapping relationship between the feature space and the rock image category, and the algorithm is tested using 100 rock thin section images from the Ordos Basin. The results show that the automatic recognition rate of rock images in different color spaces is more than 95%.

The above research on rock image recognition uses the standard rock thin section image, rather than taking the more complex and direct rock image as the research object. It is based on various more complex feature parameter extraction algorithms, and the identified rock image data are less. The current research results reduce the problem of strong subjectivity and high recognition cost in the traditional method. But, it cannot meet the requirements of geological survey personnel to quickly identify rock lithology in real time in the field. “Smartphones” are now a ubiquitous handheld communication and computing device with

multiple sensors that all workers can use anytime, anywhere. In order to get a better solution, this paper proposes a method to identify the rock image. The method is suitable for smartphones, and the recognition is fast and accurate. An application running on an android smartphone is also designed and developed in this article. Because smartphone computing and storage resources are limited, this method is based on ShuffleNet, a light convolutional neural network. Combining with the transfer learning method, the learning results of ShuffleNet on ImageNet of a large data set are transferred. These are transferred to the experimental data set in this paper, namely, the rock image data set (a total of 30 categories). After retraining, the generated rock recognition model is exported. Finally, an app was designed and developed in this paper to help the staff quickly and accurately identify the rock lithology on site. The rock recognition model needs to be deployed on android-based smartphones. The model of this paper extracted features by searching image pixels without manual operation, which reduced the influence of subjective factors. Moreover, the training process has low requirements on the rock image size, imaging distance, and light intensity. Using smartphones, which are carried by workers, lithology can be quickly and accurately identified. Compared with the traditional method, it solves the problem of the traditional method. The solved problems include strong subjectivity, high identification cost, and long cycle. It also has advantages over the analysis and feature extraction techniques based on rock slice images. For example, the method can directly identify more complex images of rocks without making thin sections. This method has the advantage of quickly and accurately identifying rock lithology in the field, which can meet the requirements of workers to identify rocks quickly and accurately.

## 2. Materials and Methods

### 2.1. Rock Recognition Model Structure Design

*2.1.1. ShuffleNet.* ShuffleNet is an extremely efficient convolution structure designed for smartphones. It was proposed by Zhang Jian and others [17]. The depthwise separable convolution and group convolution introduced by Xception and ResNeXt can coordinate the ability and computation of the model, but their pointwise convolution occupies a large amount of computation [18–21]. ShuffleNet introduces pointwise group convolution to solve this problem. It has two characteristics: pointwise group convolution and channel shuffle. Compared with the existing advanced CNN models (such as MobileNets) [22, 23], the calculation amount can be greatly reduced under the similar precision, and the parameter amount can be greatly reduced. A large number of  $1 \times 1$  convolutions consume a lot of computing resources, and pointwise group convolution helps reduce computational complexity. As shown in Figure 1, channel shuffle is to orderly disrupt the channels of each feature map to form a new feature map to solve the problem of “poor information flow” caused by group convolution.

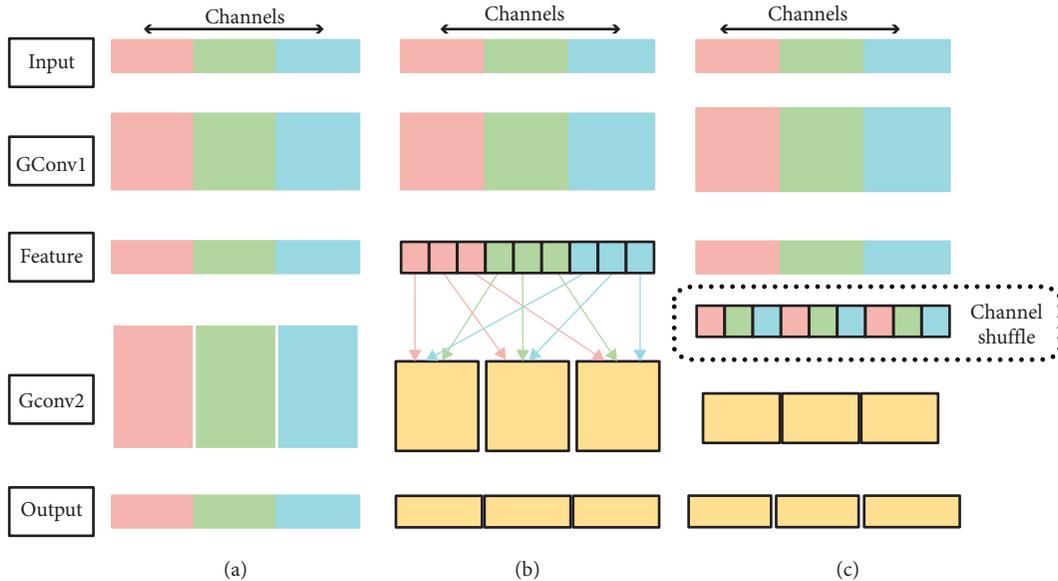


FIGURE 1: Channel shuffle with two stacked group convolutions. GConv stands for group convolution. (a) Two stacked convolution layers with the same number of groups. Each output channel only relates to the input channels within the group. No cross talk; (b) input and output channels are fully related when GConv2 takes data from different groups after GConv1; (c) an equivalent implementation to (b) using channel shuffle.

Group convolution can effectively reduce the computational cost, but the output only comes from some fixed input channels. It prevents feature exchange between channels and does not obtain the optimal representation. ShuffleNet uses channel shuffle to construct the association between the input channel and the output channel, including a convolutional layer with  $g$  groups and an output with  $g \times n$  channels. The output dimension is reshaped into  $(g, n)$ , and it is transposed and flattened as the input to the next layer. Figure 2 shows that ShuffleNet is based on channel shuffle to construct the ShuffleNet unit.

The ShuffleNet architecture is primarily built from a set of ShuffleNet units. A ShuffleNet unit consists of a  $1 \times 1$  pointwise group convolution layer and follows the channel shuffle operation layer. Under the same conditions, the calculation cost of this structure is low. The input is  $c \times h \times w$  with bottleneck channels  $m$ . ShuffleNet only requires  $hw(2cm/g + 9m)$  FLOPs, but ResNet requires  $hw(2cm + 9m^2/g)$  floating-point operations per second (FLOPs). Compared to MobileNet, the ShuffleNet model achieves an absolute 7.8% performance in ImageNet Top-1 errors at a cost of approximately 40 million floating-point operations per second (MFLOPs). Channel split operation was proposed in the ShuffleNet V2. Firstly, the input of the feature channel is divided into two branch channels. One branch remains unchanged, and the other branch is computed a  $1 \times 1$  convolution and  $3 \times 3$  depthwise separable convolution. Then, the two branch features are connected, and the channel shuffle operation is implemented. After the channel is reorganized, the next unit is repeated. The report shows that ShuffleNet V2 is about 40% faster than ShuffleNet V1 and about 16% faster than MobileNet V2. With 500 MFLOPs, ShuffleNet V2 is 58% faster than MobileNet V2 and 63% faster than ShuffleNet V1 [24, 25].

**2.1.2. Transfer Learning and Model Construction.** The transfer learning method can apply the knowledge learned from other tasks (source tasks) to the target task. This method is conducive to the construction of the mathematical model of the target task and reduces the duplication of labor and the dependence on training data of the target task [26–28]. A comparison of transfer learning and traditional machine learning is shown in Figure 3. Traditional machine learning faces different learning tasks, and even if there is a similarity between tasks, different learning systems need to be established. However, in the face of different learning tasks, transfer learning can transfer the knowledge learned from the learning system. In other words, knowledge learned in solving the source task is transferred to the learning system that solves the target task.

From the perspective of the structure and function of the deep neural network, the convolutional layer of the neural network mainly extracts features and shares parameters and reduces the number of parameters through the use of the pooling layers [27–30]. The features extracted by the network are integrated through the final fully connected layer to obtain the high-level meaning of the image features. Then, it is classified by the classifier to get the final classification result [26, 31]. In some cases, the data set is small, and the distribution is not balanced, which makes the training results overfitting. The model performs well on the training data set but performs poorly on the verification data set and test data set. Using transfer learning method can improve this problem very well.

**2.1.3. Model Structure Design.** Through transfer learning, the model parameters trained by ShuffleNet on ImageNet of the large data set are migrated. In order to help train the rock image recognition model, it was migrated to 30 classes in the

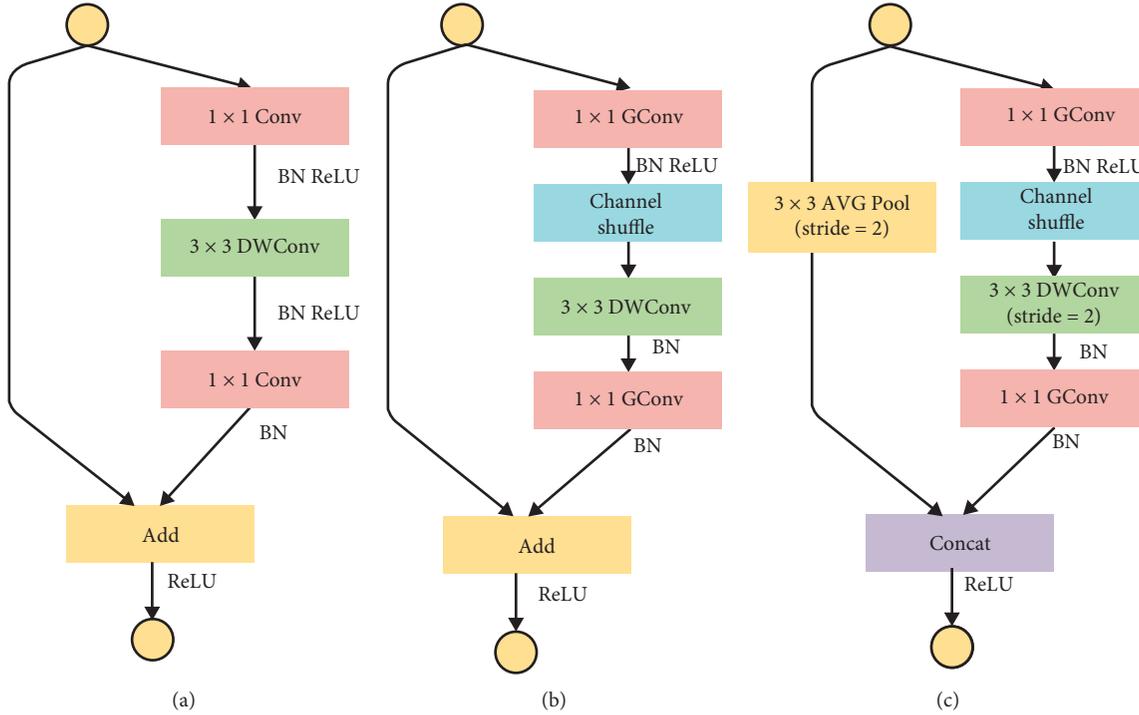


FIGURE 2: ShuffleNet units. (a) Bottleneck unit with depthwise convolution (DWConv); (b) ShuffleNet unit with pointwise group convolution (GConv) and channel shuffle; (c) ShuffleNet unit with stride = 2.

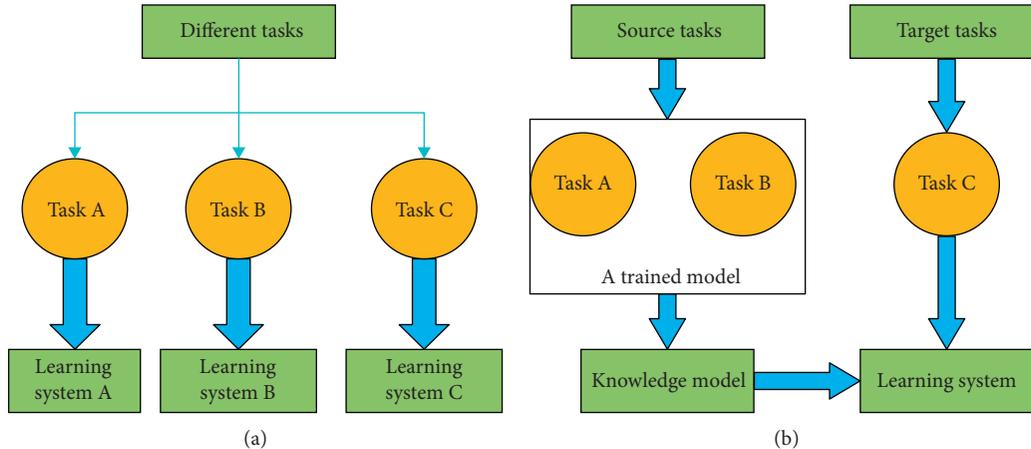


FIGURE 3: Comparison of traditional machine learning (a) and transfer learning (b).

experimental data set of the rock image. The ShuffleNet can extract valid information from images [32, 33]. The difference between rock image data sets and large data sets is relatively small. And in specialized fields, it is a small data set of fine-grained types. Therefore, the problem of rock image recognition belongs to the fine-grained classification of small data sets. This paper used the transfer learning method to perform rock image recognition. Most parameters of the network pretrained on large data sets were retained and adjusted to fit this data set. Input image resolution: 128, 160, 192, or 224px. Different sizes of input pictures will affect the classification results [34]. This article used 224 as the initial setting. The relative size of the model can be set to 1.0, 0.75,

0.50, or 0.25. This paper recommended 0.5 as the initial setting. Smaller models run significantly faster but at the expense of accuracy.

In order to deploy the trained model to the smartphone, the lightweight convolutional neural network structure ShuffleNet of 2.1.1 was used. ShuffleNet weights and parameters pretrained by ShuffleNet on the ImageNet data set were imported based on the characteristics of the rock image data set. Each convolutional layer used ReLU as the activation function. Batchnorm was used to normalize the distribution of the batch. The Softmax classifier of 2.1.4 was used for classification. A model was trained on the data set. The model structure is shown in Table 1.

TABLE 1: ShuffleNet architecture.

Layer	Output size	$K$ size	Stride	Repeat	Output channels ( $g$ groups)				
					$g=1$	$g=2$	$g=3$	$g=4$	$g=5$
Image	$224 \times 224$				3	3	3	3	3
Conv1	$112 \times 112$	$3 \times 3$	2	1	24	24	24	24	24
MaxPool	$56 \times 56$	$3 \times 3$	2	1					
Stage 2 <sup>1</sup>	$28 \times 28$		2	1	144	200	240	272	384
	$28 \times 28$		1	3	144	200	240	272	384
Stage 3			2	1	288	400	480	544	768
			1	7	288	400	480	544	768
Stage 4	$7 \times 7$		2	1	576	800	960	1088	1536
	$7 \times 7$		1	3	576	800	960	1088	1536
GlobalPool	$1 \times 1$	$7 \times 7$							
FC					1000	1000	1000	1000	1000
Complexity <sup>2</sup>					143M	140M	137M	133M	137M

**2.1.4. Softmax Regression Model.** In the multiclassification problem, the Softmax regression algorithm was adopted in the rock recognition model in this paper to map the output values for multiple neural units into  $(0, 1)$  with a total value of 1 [29, 35]. Therefore, the rock recognition model was classified as the probability of a sample being in a certain category to realize multiclassification. Let the training set consist of  $m$  labeled samples, i.e.,  $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ . The range of the category label  $y$  is  $y^{(i)} \in \{1, 2, \dots, k\}$ . Let probability  $p(y = j|x)$  denote the probability that the sample is discriminated as being in category  $j$  in the case of input  $x$ . Therefore, the output of the  $k$ -class classifier is a  $k$ -dimensional vector, and the sum of its elements is 1. Analogical logistic regression using the hypothesis function  $h_\theta(x^{(i)})$  can express the output of Softmax as

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}. \quad (1)$$

Among the terms in the equation, the input  $x$  is a vector of dimension  $m \times 1$ , and the output model parameter is a matrix of order  $m \times k$ . The training process of the model is used to find the optimal value through continuous iteration so that the predicted value approaches the actual value. The cost function of the regression model can be expressed as

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right], \quad (2)$$

where  $1\{\times\}$  is an indicative function, whose value rule is  $1\{\text{expression whose value is true}\} = 1$ ;  $1\{\text{expression whose value is false}\} = 0$ .

As for solving the parameters by minimizing  $J(\theta)$ , there is currently no closed solution to minimize  $J(\theta)$ . In this paper, iterative algorithms such as gradient descent are used

to solve the problem. The gradient formula that was obtained after taking derivatives is as follows:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m [x^{(i)} (1\{y^{(i)} = j\} - p(y^{(i)} = j | x^{(i)}; \theta))], \quad (3)$$

$\nabla_{\theta_j}$  is a vector whose  $l$  element  $\partial J(\theta) / \partial \theta_{jl}$  is the partial derivative of  $J(\theta)$  to the  $l$  component of  $\theta_j$ .

In this paper, a weight attenuation function term is added to the cost function to make it strictly convex to ensure its convergence and unique solution. The cost function is modified by adding  $\lambda/2 \sum_{i=1}^m \sum_{j=0}^k \theta_{ij}^2$ , where  $n$  represents the number of input data, and this weight attenuation term will punish excessive parameter values. The cost function was converted to the following equation:

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{i=1}^m \sum_{j=0}^k \theta_{ij}^2. \quad (4)$$

After adding the weight attenuation term ( $\lambda > 0$ ), the cost function became a strict convex function so that a unique solution can be guaranteed, and since  $J(\theta)$  is a convex function, the gradient descent method can guarantee convergence to the global optimal solution. Divide the data set into 3 parts: training set, validation set, and test set, take the number in  $\lambda$  from small to large, and then learn the model parameters on the training set, calculate the verification set error on the cross-validation set, and select the model with the smallest error, that is, choose  $\lambda$ . Finally, the evaluation is performed on the test set to obtain the best  $\lambda$  value. In order to use the optimization algorithm, the derivative of this new function  $J(\theta)$  is required:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m [x^{(i)} (1\{y^{(i)} = j\} - p(y^{(i)} = j | x^{(i)}; \theta))] + \lambda \theta. \quad (5)$$

An available Softmax regression model can be achieved by minimizing  $J(\theta)$ .

## 2.2. Model Training

*2.2.1. Data Set and Data Preprocessing.* Geological survey workers use fresh rocks by using a smartphone as an experiment in a data set of rock images. These rocks are generally rock foam rhyolite of different lithologies and dark gray stromatolite almond-like rough rocks. For example, there are 30 kinds of light gray rhyolite, purple red tuff, gray black obsidian, purple gray amphibolic rhyolite, and potassium gray white rock data. These images came from multiple locations in East China, with sizes between 3M and 6M. In this paper, the size of each image is compressed to  $224 \times 224$  pixels on the condition of ensuring accuracy. Figure 4 is a sample map of the rock.

30 different kinds of rocks were collected, and a total of 3,795 images were taken. According to the ratio of 8:1:1, images were randomly selected from rock samples as the training data set, verification data set, and test data set. That is, there are 3,046 graphs of the training set, 381 graphs of the verification set, and 368 graphs of the test set. The detailed data distribution is shown in Figure 5.

It can be seen from the observation of the data set that the number of all types of data was unbalanced, and the number of pictures of some categories was very less. Methods such as rotation, flipping, cutting, and adjusting light and shade were used to randomly expand the training data set to improve the training performance.

*2.2.2. Training Model.* The transfer learning method was used to train the rock recognition model in the TensorFlow framework on the PC. The ShuffleNet network structure was built using the Python programming language, and the parameters pretrained by ShuffleNet on the ImageNet data set were imported. Experiments were evaluated on Core I9 series CPU, 32G RAM, NVIDIA GeForce GTX Titan Z 12G GPU, Linux OS PC. In the training process, the default iteration step number was 3600, and the learning rate was 0.008. The activation function was ReLU. During each iteration, 50 images were randomly selected from the data set for training, and 15 images were randomly selected for cross-validation. Softmax was used as a classifier to classify, and the optimization function used a method of stochastic gradient descent. Training accuracy refers to the percentage of accurate classification of currently trained images, while verification accuracy refers to the percentage of accurate classification of randomly selected images. Cross-entropy displays the learning effect of the model training process. The smaller the value, the better the learning effect.

As can be seen from Figure 6, the loss converges after the 160th iteration. After the 640th iteration, it remained stable, and the loss was close to zero. And at this point, the training accuracy is close to 100%. The accuracy of the verification set is slightly tight. After 3,600 iterations, the accuracy of the rock recognition model on the training set approaches 100%, and the loss is only 0.0004; the accuracy on the verification data set reached 97.65%, and the loss is only 0.1052. According to the training accuracy, verification accuracy, and cross-entropy changes, it can be seen that the training effect of the model is relatively ideal.

*2.3. Accuracy and Run Time Comparison.* This paper shows the superiority of the rock recognition model which is based on the combination of the ShuffleNet convolutional neural network and the transfer learning method. On the personal computer, the precision and running of the rock recognition model based on ShuffleNet were compared with the MobileNet model, the SqueezeNet model, and the standard convolutional network ResNet50 model. Originally designed for mobile and embedded visual applications, MobileNets are built primarily from the deep separable convolution operation, which decomposes standard convolution into deep convolution and point-by-point convolution. MobileNets apply a single filter to each input channel and then combine the output with linear combinations through point-by-point convolution. MobileNet V2 is proposed for further improvement. It is constructed by inversion residual and linear bottleneck technique, which can reduce the number of parameters and the loss of activation operation. Combined with the single-shot detector lite used for object detection, it was reported that MobileNet V2 was 35% faster than MobileNet V1, with 20 times less computation and 10 times fewer parameters than YOLO V2 [36]. SqueezeNet proposes to maintain precision with a small number of parameters, and its core structure is a new type of component Fire module. There are three main strategies for building Fire module [37, 38]. First, the  $3 \times 3$  filter was replaced with a  $1 \times 1$  filter. Second, the number of input channels of the filter was reduced. Finally, the network sampling was delayed. The evaluation results showed that the structural parameters of SqueezeNet were 50 times less than original AlexNet and maintained the horizontal accuracy of AlexNet on Imagenet.

*2.4. Software Deployment.* In this study, the trained rock recognition model was deployed on smartphones or embedded products. The significance of rock identification lies in the fact that geological investigators use smartphones for recognition on the work site instead of huge servers in the laboratory. Recognition in the laboratory is not a bad option. But, if workers take rock samples back to the laboratory to make rock thin sections, the recognition period and cost will increase. Many applications are often very sensitive to the response time of a program; even small latency in service response can have a significant impact on the user. Today, more and more applications provide core functionality through deep learning models. Whether people are deploying models to the cloud or to smartphones, low-latency reasoning is becoming increasingly important. One way to solve this problem is to perform model inference on a high-performance cloud server. Also, the input and output models are transferred between the client and the server. However, this solution brings many problems, such as high computing costs, massive data migration through mobile networks, user privacy, and increased latency. The top-level compressed ShuffleNet model takes an alternative approach to these scenarios and requires less resources to perform reasoning. This section describes the process of deploying the ShuffleNet model on a smartphone.

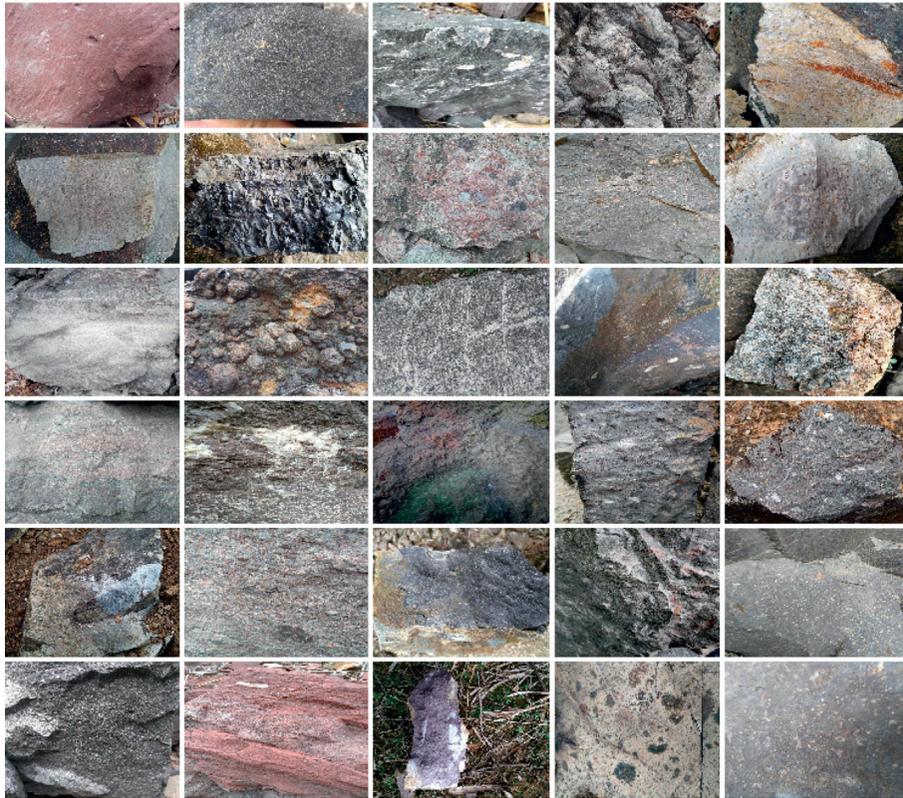


FIGURE 4: Rock sample data.

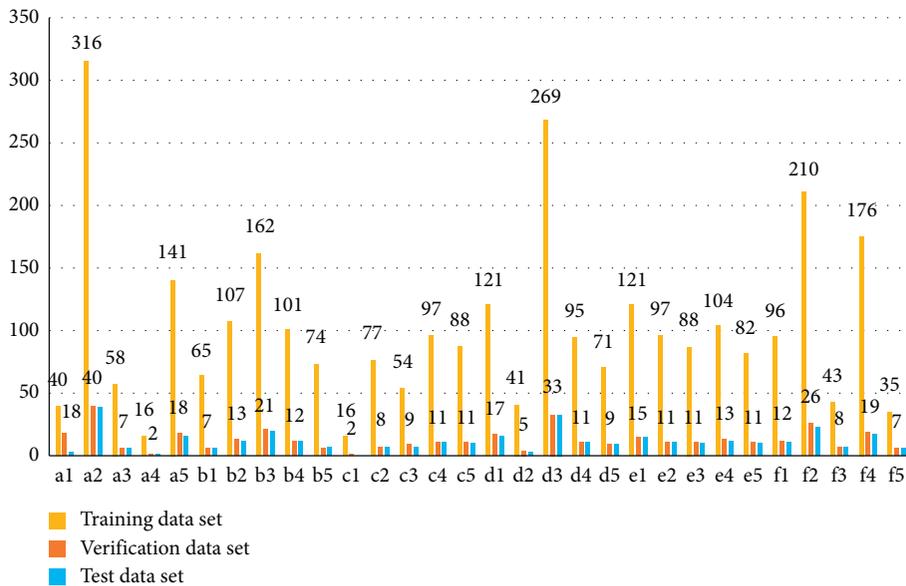


FIGURE 5: Data distribution.

The rock recognition model is trained on a PC server. This paper uses the TensorFlow framework to run properly on Linux. However, this cannot be done directly on the smartphone and requires some necessary conversions and deployments. The CNN model on Linux needs to be converted to the (.pb) format and deployed on Android smartphones. In order to implement the solution of

identifying rock lithology in the field, this paper developed an application running on Android smartphones. In addition, Huawei, Samsung, and Oppo, three common smartphones in the market, were selected as the experimental platform. The interfaces of the application are shown in Figure 7.

The application is written in the Java programming language and runs on an Android smartphone (Android

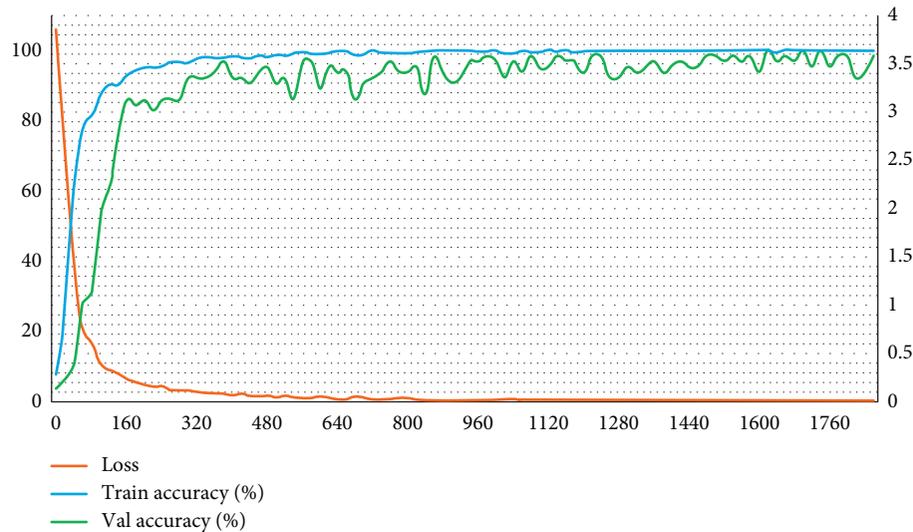


FIGURE 6: Training set accuracy, verification set accuracy, and loss during training.

4.4 operating system or higher). The operating memory of the phone should be greater than 4 GB, and the storage capacity should be 32 GB or more. The application can load and run the trained CNN model under the TensorFlow framework [39, 40]. It can identify not only the rock image captured by the smartphone in the field but also the rock photos stored in the gallery of the mobile phone. Software can output the recognition results (the type and nature of the rock, the accuracy of the recognition, and the time of execution) to the interface. The Huawei P20 mobile phone is used for the accuracy test of the model on the smartphone. The built-in main camera has a focal length of 3.95 mm and a resolution of  $2244 \times 1080$  pixels, which is supported by most smartphones. Samsung Galaxy A8 s and Oppo R17 with a 16 megapixel rear camera (Android 7.0) are used to illustrate the performance of this app on other phones.

Operating the application is simple and convenient. Install the application on the smartphone carried by the investigator. Open the software at work. Then, click “Camera” to enter the camera photographing interface. Then, point the camera at the rock and click “photographing.” After this series of steps, the captured image is loaded into the interface to be recognized. Then, click “recognize,” and the recognition result (including the type and lithology of the rock, the recognition accuracy, and the recognition time) is displayed on the interface in less than 1 second.

**2.4.1. Software Development Platform.** On Windows 10, build the application development environment based on Android Studio 3.3, Android SDK (Java Development Kit), Java JDK 8 (Java Development Kit), TensorFlow Lite Development Kit (you can migrate the model trained under the TensorFlow framework to Android smartphone), and ADT (Android development tools). The application is suitable for the operating system of Android 6.0 and above.

**2.4.2. Software Performance and Technical Overview.** This software can be used by all types of Android smartphone users. It implements a friendly graphical user interface and features linear execution. In this interface, only the rock identification results (type, lithology, and accuracy) and the identification time are shown. It also allows the user to use the application without knowing its internal performance. The technical system of the application program is basically composed of two parts. One is the underlying part composed of the TensorFlow Lite development interface, and the other is the Android application layer composed of the Android native development interface (API) [41, 42]. TensorFlow Lite can deploy the CNN model trained under the TensorFlow framework to Android smartphones. According to the principle described in Section 2.1, TensorFlow Lite provides a Jar package written in Java and (.so) format dynamic library written in C++. The latter provides APIs for operational models such as functions for reading models, recognition functions, and output functions. The “Android native API” implements the main parts of the main application and the graphical user interface. It is responsible for coordinating tasks, invoking Android camera sensors to capture rock images and correctly store results. When rock images are captured in the “Android application layer” and needed to be identified and analyzed, a request was made for “TensorFlow Lite underlying part.” However, direct communication between the two parts is not feasible. So, this paper needs to use the Java Native Interface (JNI) to allow this interaction. The “Android application part” calls the required functionality through JNI, which is actually responsible for executing the C++ library and returning the results.

### 3. Results

**3.1. The Accuracy and Time of the Rock Recognition Model Tested on the Smartphone.** The purpose of this study is to facilitate geologists. The idea is for them to use smartphones

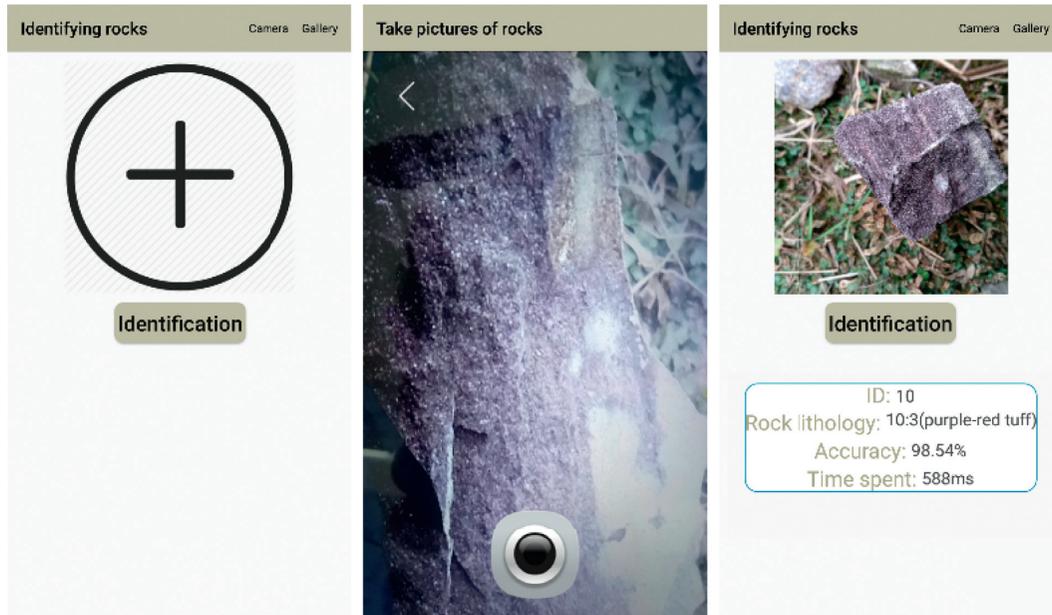


FIGURE 7: Application operator interface.

to quickly and accurately identify the types and lithology of rocks in the wild. Therefore, the accuracy and running time of identification are very important. The test data set and the (.pb) format model which has completed the training and the (.txt) format label file containing rock information are imported into the SD card of the Huawei smartphone, respectively. When the application in 2.2 is run, the rock recognition model is automatically loaded. Then, the test data set in the smartphone is read, and the recognition result of the test set is obtained. As shown in Figure 8, the accuracy of the model is represented by the confusion matrix.

Among them, the row value of the matrix is the true value. The column value is the predicted value, and the accuracy of the whole test set is 95.30%. And the accuracy of the single image is above 96%, accounting for 95% of the test data set.

The recognition time distribution of the single image of the test data set is shown in Figure 9. Among them, the average recognition time of the single image is 786 milliseconds. The maximum is 1,045 milliseconds. And the minimum is 452 milliseconds. The boxplot has no outliers, indicating stable model recognition.

**3.2. Correlative Experiments and Analysis.** A rock recognition model based on the ShuffleNet convolutional neural network combined with the transfer learning method was presented in this paper. In order to verify the superiority of this model, the same training data set and validation data set were used in this paper. And the accuracy and running time of this model were contrasted with other CNN models (MobileNet and SqueezeNet).

**3.2.1. Compared with Other CNN Models.** The training based on different batch sizes was evaluated at 8, 16, 32, and

48, respectively. Figure 10 shows the relationship between model accuracy and training period. This section takes advantage of the accuracy of the rock type and lithology. The data and other parameter settings were the same as the experimental data provided in Section 2.2.

All models converged after about 35 training epochs. The 32-batch training model achieved better performance, about 5% better than the other models. SqueezeNet was more stable and smooth during training, while MobileNet and ShuffleNet fluctuated more. It is reasonable to find that the calculation of gradient descent direction was more accurate and milder for larger batch sizes during model training. Smaller batch sizes resulted in more randomness and made it harder to achieve optimal performance.

**3.2.2. Execution Time Evaluation.** Different CNN models contain layers of various depths and widths, number of filters, and size and shape of filters, which lead to different structures, parameters, and complexity. In this paper, rock recognition models based on ShuffleNet training are compared with models based on other convolutional neural network training. The following results were obtained by evaluating the running time of different models. Running time of different CNN models was evaluated. Training and testing time of MobileNet, ShuffleNet, SqueezeNet, and ResNet50 models with various batch sizes (8, 16, 32, and 48) were evaluated. Table 2 gives the experiment results.

Using batch sizes 8, 16, 32, and 48 during each iteration, the MobileNet model had the longest training time, and these training times were 1.265 s, 2.364 s, 4.728 s, and 8.512 s, respectively. The ShuffleNet model had the shortest training time, about 75% of that of MobileNet. For test time, ShuffleNet took the shortest amount of time, with a time of 0.125 s. Comparing with the ResNet50 model, the running efficiency was greatly improved with compressed CNN

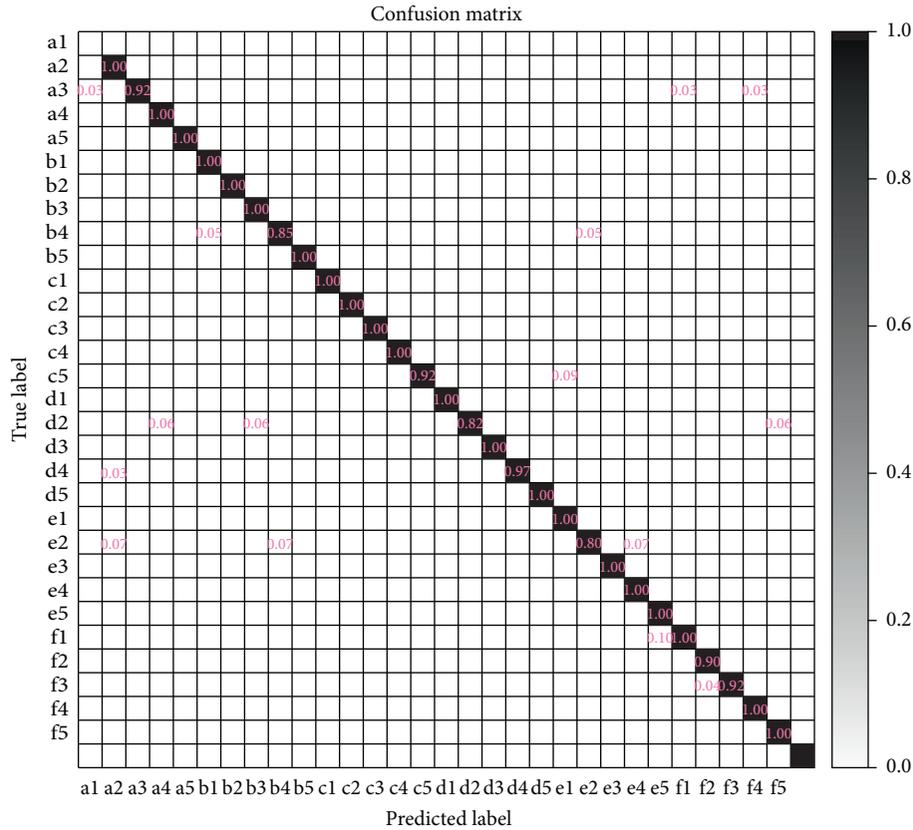


FIGURE 8: Confusion matrix.

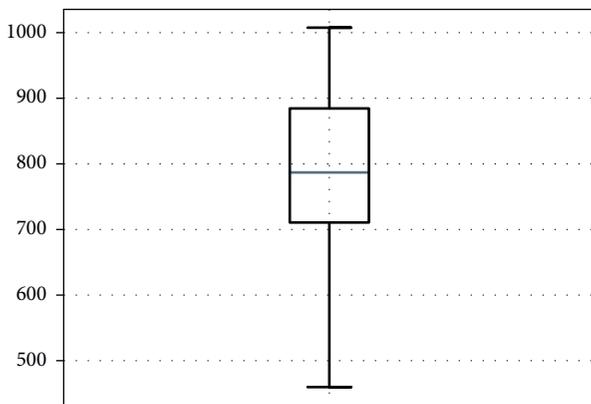


FIGURE 9: Recognition time of a rock image on the smartphone.

models. For the size of space occupied by the model, MobileNet, ShuffleNet, and SqueezeNet required 34.5M, 18.2M, and 25M, respectively, but 219.4M is needed for the ResNet50 model. The experimental results also show that the ShuffleNet model is efficient and occupies less space. It is 7 times faster than the standard convolutional network ResNet50 model and takes up 12 times less space.

**3.3. Comparing the Recognition Times Using Different Android Smartphones.** The compressed CNN model was deployed on the Android smartphone, and its performance was tested. After the model was converted, ShuffleNet, MobileNet, and

SqueezeNet files were, respectively, 15.2 MB, 8.4 MB, and 42.8 MB. Table 3 shows the model test results for the selected Android smartphone. Huawei, Samsung, and Oppo phones were used for testing, and the processors of the three phones are Kylin970, Qualcomm710, and Qualcomm670. The results showed that smartphones were very efficient and could perform operations in 0.5 seconds, enabling real-time applications. The Huawei phone achieved the best performance, taking 0.283 seconds to execute the model due to the neural network processing unit contained in it. Three models were deployed on the same smartphone. There were three different types of smartphones. As shown in Table 3, the model of ShuffleNet achieved the best performance.

**3.4. The Advantages of the Presented Method.** The rock identification method in this paper is compared with the traditional method and the method based on rock slice image processing. Table 4 shows the advantages of the presented method. The model of this paper can quickly identify the types and properties of rocks on the condition that the accuracy requirements are met. The presented method can quickly get the recognition results in less than 1 second after taking photos in the field. And there is no need to make rock flakes to reduce the cost of identification. The experimental results show that the convolutional network model has obvious performance in model compression and computation. It is suitable for rapid and accurate recognition of rock lithology under field offline conditions.

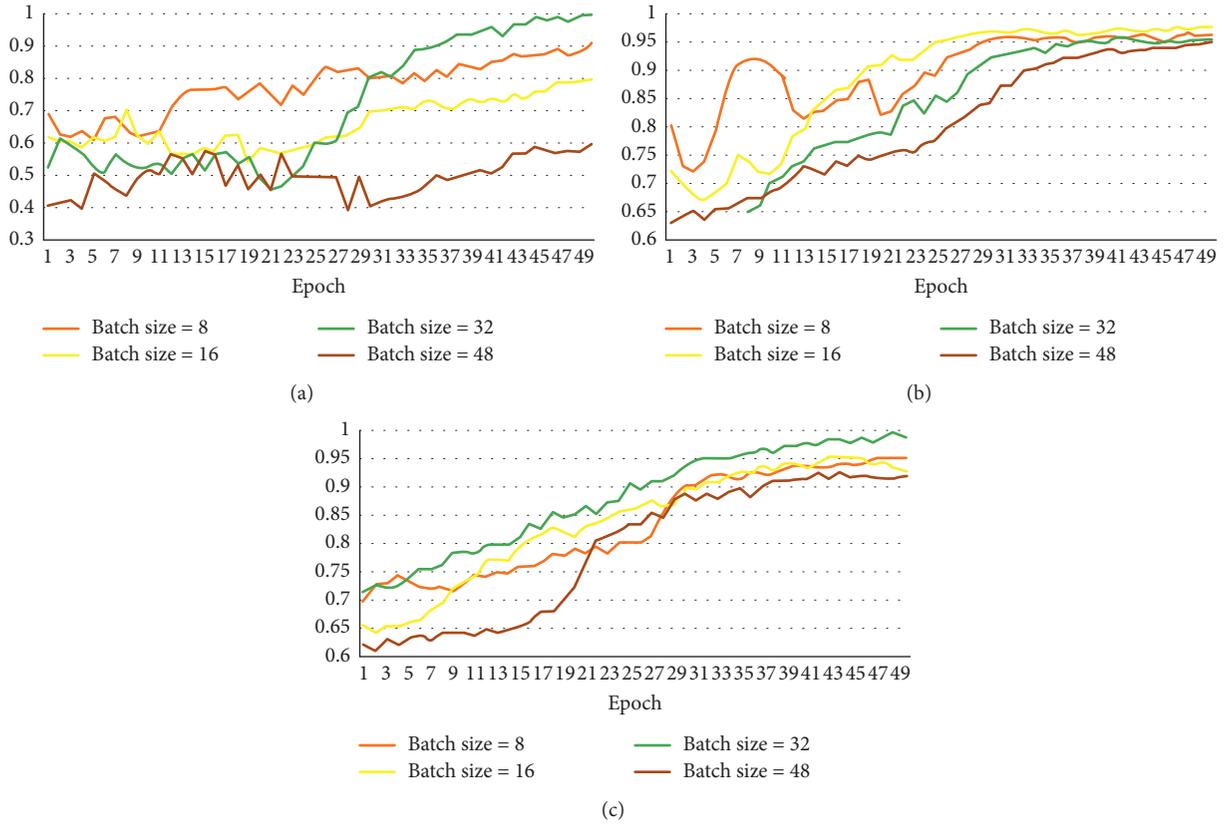


FIGURE 10: Model accuracy versus training epoch on various batch sizes: (a) result on the MobileNet model; (b, c) result on SqueezeNet and ShuffleNet models, respectively. Orange, yellow, green, and wine curves represent batch sizes equal 8, 16, 32, and 48, respectively.

TABLE 2: Execution time (millisecond).

Execution time	Batch size	MobileNet	ShuffleNet	SqueezeNet	ResNet50
Training time	8	657	158	553	1465
	16	1182	212	995	2935
	32	2364	424	1770	5048
	48	4256	954	2655	8993
Testing time	X	249	125	192	915
Space occupation	X	34.5M	18.2M	25M	219.4M

TABLE 3: Execution time on Android smartphones (millisecond).

CNN models	Oppo R17	Samsung Galaxy A8s	Huawei P20	Our PC server
MobileNet	1145	531	417	249
ShuffleNet	457	308	283	125
SqueezeNet	621	367	339	192

### 4. Discussion

In this paper, lightweight convolutional neural network ShuffleNet was used to solve the problem of recognizing types and lithology of rocks in the field. The pretraining model of ShuffleNet was fine-tuned in combination with the transfer learning method, and then the model was retrained on the rock image data set in this paper. Finally, this paper developed an intelligent program for quickly identifying rocks for geological survey. This program enabled effective

recognition of 30 types of rocks such as granite, rhyolite, tuff, and breccia before deploying trained rock models to Android smartphones. In this paper, the accuracy of the recognition model reached 97.65% in the verification data set of PC. And the accuracy of the recognition model on the test data set of the smartphone was 95.30%. The average recognition time of a single rock image was 786 milliseconds. The model size was only 18.2 MB. For the same rock model, there was no significant difference in the results of different smartphone recognition. The model extracted features by

TABLE 4: The presented method is compared with traditional recognition methods and feature extraction methods based on the rock thin section.

Recognition methods	Recognition period	Recognition process and cost estimate	Whether it can be applied in the field	Recognition accuracy
The presented method	Real-time recognition	Use a smartphone to take pictures of the rocks directly. Approximately \$300	Can be recognized in the field	Meet the accuracy of the survey
Traditional recognition methods	More than 2 days	Collect rock specimens, make rock thin sections, and observe the thin sections under the microscope by professionals. Costs over \$850	Unable to recognize in the field	Meet the accuracy of the survey
The method for analyzing and feature extraction based on the rock thin section	More than 2 days	Collect rock samples, make rock thin sections, and recognize in PC. Costs over \$850	Unable to recognize in the field	Meet the accuracy of the survey

searching image pixel points without manual operation, thus reducing the influence of subjective factors. Compared with the recognition of rocks using the technique of rock thin section image processing [3, 43], the presented method has lower requirements on the size, imaging distance, and light intensity of the rock image. In this paper, the rock recognition model trained by ShuffleNet was compared with MobileNet and SqueezeNet training models, respectively, in terms of accuracy and running time. It was found that the rock recognition model based on ShuffleNet training has many advantages. For example, it can effectively reduce model parameters, compress model size, improve model calculation speed, and shorten model running time. This method had short recognition time and high accuracy and was suitable for fast and accurate recognition of rock images under offline conditions in the field. Based on ShuffleNet's lightweight convolutional neural network, the characteristics of rocks were effectively identified in the image. Through the tests on PC and smartphones, there was no wrong situation, which fully proved the robustness and generalization ability of the model.

The greatest contribution of this paper was to provide a solution for geological survey to quickly and accurately recognize rock lithology. Traditional recognition methods need not only collection of fresh rock samples to make rock thin section but also knowledgeable or experienced professionals to recognize the rock type and structure parameters under the microscope. The traditional method has strong subjectivity, long period, and high difficulty in the field. Therefore, the traditional identification method requires the observer to have very rich geological knowledge and experience [42–44]. At present, it was found that most of the rock deep learning recognition techniques are used to identify rock slices. The same is true for image processing techniques. Workers need to collect rock samples and go back to the office or laboratory to make the rock thin section. The recognition accuracy can meet the requirements of professional standards. But, the biggest disadvantage is that the research results of rock recognition cannot be applied to the field. And workers cannot use the research results of rock recognition to quickly and accurately identify the rock in the field [6, 45, 46]. This paper used ShuffleNet combined with the transfer learning method to train the rock recognition model. There was no need to make the rock thin section.

Geological investigators use smartphones which they carried as tools to photograph rock images in the field.

The presented method also has some limitations. The size of sample data has a crucial influence on the recognition effect of the deep learning model. When the number of images of a certain type of the rock is less, its features will be submerged, leading to poor recognition effect. It is difficult to find similar rock images with low probability of classification and recognition. Therefore, the probability of identifying such rock images is low. In this paper, the original training data set is expanded after cutting the rock image. Then, a new classification recognition model is established by training. The second test was made by classifying and recognizing the rock images with low probability. For granite, which contains a lot of minerals and has a wide range of variation in its content, the recognition and classification effect in the model are poor. Because granite mainly consists of feldspar, quartz, and black and white mica, the mineral composition of different varieties is not the same. And there may be pyroxene and amphibole, so the image features are complex, and the difficulty of recognition is increased. In addition, there were 30 types of rocks identified in this paper, and more types and quantities of training samples were needed. The model in this paper was compared with the training models of MobileNet and SqueezeNet. Experimental results show that the ShuffleNet-based rock recognition model has advantages in precision and running time. The comparison involves fewer models and requires more lightweight compression models. Adding more models is to choose a better solution in terms of precision and time.

## 5. Conclusions

Recognition of rock types and lithology are an important part of geological survey. In this research, ShuffleNet, a lightweight convolutional neural network designed for smartphones, was used to recognize the types and lithology of rocks. The transfer learning method was used to train the rock recognition model on the PC, and the trained model was deployed on the smartphone. This paper designed and developed an application program that runs on a smartphone. This application is not only simple in operation but also highly accurate in rock recognition. This paper solved

the problems of long recognition period and high cost in traditional recognition methods. It also makes up for the defect that the methods based on the image processing and feature extraction of rock thin section cannot recognize the rock quickly and accurately in the field. Geological investigators can quickly and accurately identify rocks by using their smartphones in the field, which is of great help to geological surveys. In the future, this paper needs to compare the rock recognition model based on ShuffleNet with more models trained by the lightweight convolutional neural network. In order to improve the accuracy and efficiency of the method, more different kinds of rock training samples were added.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Acknowledgments

The authors thank other team members for their help with the experiment. This research was jointly supported by Geological Survey Projects of China Geological Survey (DD20190416).

### References

- [1] H.-J. Massonne, H.-J. Bernhardt, D. Dettmar, E. Kessler, O. Medenbach, and T. Westphal, "Simple identification and quantification of microdiamonds in rock thin-sections," *European Journal of Mineralogy*, vol. 10, no. 3, pp. 497–504, 1998.
- [2] F. S. Reed and J. L. Mergner, "Preparation of rock thin sections," *American Mineralogist*, vol. 38, pp. 1184–1203, 1953.
- [3] C. A. Perez, J. A. Saravia, C. F. Navarro, D. A. Schulz, C. M. Aravena, and F. J. Galdames, "Rock lithological classification using multi-scale Gabor features from sub-images, and voting with rock contour information," *International Journal of Mineral Processing*, vol. 144, pp. 56–64, 2015.
- [4] L. Yu, A. Porwal, E.-J. Holden, and M. C. Dentith, "Towards automatic lithological classification from remote sensing data using support vector machines," *Computers & Geosciences*, vol. 45, pp. 229–239, 2012.
- [5] L. Lepistö, I. Kunttu, and A. J. E. Visa, "Rock image classification using color features in Gabor space," *Journal of Electronic Imaging*, vol. 14, no. 4, Article ID 040503, 2005.
- [6] A. K. Patel and S. Chatterjee, "Computer vision-based limestone rock-type classification using probabilistic neural network," *Geoscience Frontiers*, vol. 7, no. 1, pp. 53–60, 2016.
- [7] N. Singh, T. N. Singh, A. Tiwary, and K. M. Sarkar, "Textural identification of basaltic rock mass using image processing and neural network," *Computational Geosciences*, vol. 14, no. 2, pp. 301–310, 2010.
- [8] G. Cheng and W. Guo, "Rock images classification by using deep convolution neural network," *Journal of Physics: Conference Series*, vol. 887, Article ID 012089, 2017.
- [9] P. Y. Zhang, J. M. Sun, Y. J. Jiang, and J. S. Gao, "Deep learning method for lithology identification from borehole images," in *Proceedings of the 79th EAGE Conference and Exhibition 2017*, Paris, France, June 2017.
- [10] S. Rinnen, C. Stroth, A. Rißé, C. Osterreich-Henning, and H. F. Arlinghaus, "Characterization and identification of minerals in rocks by ToF-SIMS and principal component analysis," *Applied Surface Science*, vol. 349, pp. 622–628, 2015.
- [11] H. Izadi, J. Sadri, and M. Bayati, "An intelligent system for mineral identification in thin sections based on a cascade approach," *Computers & Geosciences*, vol. 99, pp. 37–49, 2017.
- [12] Y. Zhang, M. Li, and S. Han, "Automatic identification and classification in lithology based on deep learning in rock images," *Acta Petrologica Sinica*, vol. 34, no. 2, pp. 333–342, 2018.
- [13] N. Li, H. Hao, Q. Gu, D. Wang, and X. Hu, "A transfer learning method for automatic identification of sandstone microscopic images," *Computers & Geosciences*, vol. 103, pp. 111–121, 2017.
- [14] G. Cheng, Q. Yue, and X. Qiang, "Research on feasibility of convolution neural networks for rock thin sections image retrieval," in *Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 2539–2542, Xi'an, China, May 2018.
- [15] R. Marmo, S. Amodio, R. Tagliaferri, V. Ferreri, and G. Longo, "Textural identification of carbonate rocks by image processing and neural network: methodology proposal and examples," *Computers & Geosciences*, vol. 31, no. 5, pp. 649–659, 2005.
- [16] C. Guo and Y. Liu, "Recognition of rock images based on multiple color spaces," *Science Technology and Engineering*, vol. 14, pp. 247–251+255, 2014.
- [17] Zhang X., Zhou X., Lin M., Sun J. S. N., An extremely efficient convolutional neural network for mobile devices, arXiv:1707.01083 [cs] 2017.
- [18] F. Chollet, "Deep learning with depthwise separable convolutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, Honolulu, HI, USA, July 2017.
- [19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Springer International Publishing, Berlin, Germany, pp. 818–833, 2014.
- [20] F. Bordignon, L. P. D. Figueiredo, R. Exterkoetter, B. B. Rodrigues, and M. D. Correia, "Deep Learning for Grain Size and Porosity Distributions Estimation on micro-CT Images," in *Proceedings of the 16th International Congress of the Brazilian Geophysical Society & Expogef*, Rio de Janeiro, Brazil, August 2019.
- [21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, February 2017.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [23] Han S., Mao H., Dally W. J., Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding, arXiv:1510.00149 [cs] 2015.
- [24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear

- bottlenecks,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.
- [25] Niu W., Ma X., Wang Y., Ren B, 26 ms inference time for ResNet-50: towards real-time execution of all DNNs on smartphone, arXiv:1905.00571 [cs, stat] 2019.
- [26] Y. Ganin, E. Ustinova, H. Ajakan et al., “Domain-adversarial training of neural networks,” in *Domain Adaptation in Computer Vision Applications*, G. Csurka, Ed., Springer International Publishing, Cham, Switzerland, pp. 189–209, 2017.
- [27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 3320–3328, Curran Associates, Inc., New York, NY, USA, 2014.
- [28] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710, Salt Lake City, UT, USA, June 2018.
- [29] Jang E., Gu S., Poole B., Categorical reparameterization with gumbel-softmax, arXiv:1611.01144 [cs, stat] 2016.
- [30] F. Zang and J. Zhang, “Softmax discriminant classifier,” in *Proceedings of the 2011 Third International Conference on Multimedia Information Networking and Security*, pp. 16–19, Shanghai, China, November 2011.
- [31] A. Caliskan, M. E. Yuksel, H. Badem, and A. Basturk, “Performance improvement of deep neural network classifiers by a simple training strategy,” *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 14–23, 2018.
- [32] Xie X., Zhou Y., Kung S.-Y., H. G. C.: hierarchical group convolution for highly efficient neural network, arXiv: 1906.03657 [cs] 2019.
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105, Curran Associates, Inc., New York, NY, USA, 2012.
- [34] E. Zawadzka-Gosk, K. Wolk, and W. Czarnowski, “Deep learning in state-of-the-art image classification exceeding 99% accuracy,” in *Advances in Intelligent Systems and Computing*, pp. 946–957, Springer, Cham, Switzerland, 2019.
- [35] S. Tao, T. Zhang, J. Yang, X. Wang, and W. Lu, “Bearing fault diagnosis method based on stacked autoencoder and softmax regression,” in *Proceedings of the 2015 34th Chinese Control Conference (CCC)*, pp. 6331–6335, Hangzhou, China, July 2015.
- [36] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, July 2017.
- [37] Shafiee M. J., Li F., Chwyl B. W. A., SquishedNets, Squishing SqueezeNet further for edge device scenarios via deep evolutionary synthesis, arXiv:1711.07459, 2017.
- [38] A. G. Santos, C. O. De Souza, C. Zanchettin, D. Macedo, A. L. I. Oliveira, and T. Ludermit, “Reducing SqueezeNet storage size with depthwise separable convolutions,” in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, Rio de Janeiro, Brazil, October 2018.
- [39] M. Abadi, P. Barham, J. Chen et al., “TensorFlow: a system for large-scale machine learning,” in *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming—ICFP 2016*, pp. 265–283, Nara Japan, September 2016.
- [40] M. A. S. Adhiwibawa, M. R. Ariyanto, A. Struck, K. R. Prilianti, and T. H. P. Brotosudarmo, “Convolutional neural network in image analysis for determination of mangrove species,” in *Proceedings of the Third International Seminar on Photonics, Optics, and Its Applications (ISPhOA 2018)*, Surabaya, Indonesia, August 2019.
- [41] H. Liu, X. Ma, M. Tao et al., “A plant leaf geometric parameter measurement system based on the android platform,” *Sensors*, vol. 19, no. 8, p. 1872, 2019.
- [42] J. Kang, J. Lee, and D.-S. Eom, “Smartphone-based traveled distance estimation using individual walking patterns for indoor localization,” *Sensors*, vol. 18, no. 9, p. 3149, 2018.
- [43] S. Karimpouli and P. Tahmasebi, “Segmentation of digital rock images using deep convolutional autoencoder networks,” *Computers & Geosciences*, vol. 126, pp. 142–150, 2019.
- [44] L. Shu, G. R. Osinski, K. McIsaac, and D. Wang, “An automatic methodology for analyzing sorting level of rock particles,” *Computers & Geosciences*, vol. 120, pp. 97–104, 2018.
- [45] S. Aligholi, R. Khajavi, and M. Razmara, “Automated mineral identification algorithm using optical properties of crystals,” *Computers & Geosciences*, vol. 85, pp. 175–183, 2015.
- [46] F. J. Galdames, C. A. Perez, P. A. Estévez, and M. Adams, “Classification of rock lithology by laser range 3D and color images,” *International Journal of Mineral Processing*, vol. 160, pp. 47–57, 2017.