

Article

An Optimization Method for Lightweight Rock Classification Models: Transferred Rich Fine-Grained Knowledge

Mingshuo Ma ¹, Zhiming Gui ¹ , Zhenji Gao ^{2,3,*} and Bin Wang ^{2,3}
¹ Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; mamingshuo@emails.bjut.edu.cn (M.M.); zmgui@bjut.edu.cn (Z.G.)

² Integrated Natural Resources Survey Center, CGS, No. 55 Yard, Honglian South Road, Xicheng District, Beijing 100055, China

³ Technology Innovation Center of Geological Information Engineering of Ministry of Natural Resources, Beijing 100055, China

* Correspondence: gzhenji@mail.cgs.gov.cn

Abstract: Rock image classification represents a challenging fine-grained image classification task characterized by subtle differences among closely related rock categories. Current contrastive learning methods prevalently utilized in fine-grained image classification restrict the model's capacity to discern critical features contrastively from image pairs, and are typically too large for deployment on mobile devices used for in situ rock identification. In this work, we introduce an innovative and compact model generation framework anchored by the design of a Feature Positioning Comparison Network (FPCN). The FPCN facilitates interaction between feature vectors from localized regions within image pairs, capturing both shared and distinctive features. Further, it accommodates the variable scales of objects depicted in images, which correspond to differing quantities of inherent object information, directing the network's attention to additional contextual details based on object size variability. Leveraging knowledge distillation, the architecture is streamlined, with a focus on nuanced information at activation boundaries to master the precise fine-grained decision boundaries, thereby enhancing the small model's accuracy. Empirical evidence demonstrates that our proposed method based on FPCN improves the classification accuracy mobile lightweight models by nearly 2% while maintaining the same time and space consumption.

Keywords: rock images classification; contrastive learning; knowledge distillation



Citation: Ma, M.; Gui, Z.; Gao, Z.; Wang, B. An Optimization Method for Lightweight Rock Classification Models: Transferred Rich Fine-Grained Knowledge. *Sensors* **2024**, *24*, 4127. <https://doi.org/10.3390/s24134127>

Academic Editors: Kuihua Wang and Juntao Wu

Received: 25 April 2024

Revised: 21 June 2024

Accepted: 24 June 2024

Published: 25 June 2024



Copyright: © 2024 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/>).

1. Introduction

The classification of rock types is an essential aspect of geological studies, and the precise and efficient categorization of rock images is of considerable importance. This classification is regarded as a task of fine-grained visual categorization [1].

Deep learning-based computer vision technologies augment image sensors by processing and interpreting visual data, enabling machines to respond to their environment in a manner akin to human vision. Advanced algorithms [2] extract valuable rock feature information from sensor data, then enhance image quality to aid in image recognition and classification, effectively extending the utility of image sensors beyond simple image capture.

Despite marked advancements in general image classification through convolutional neural networks (CNNs) in recent years [3], the fine-grained classification of rock images remains a significant challenge owing to the pronounced similarities among subclasses.

Recent advances have been made in the automated classification of rock images. Chatterjee et al. [4] extracted textural features from rock images and applied a Support Vector Machine (SVM)-based approach for regression prediction utilizing these features. Liang et al. [2] employed CNN-based methods, while Guojian [5] introduced a rock slice

image classification technique using residual networks. Pascual et al. [6] utilized a three-layer CNN architecture to classify rock images, resulting in enhanced recognition precision. To bolster the learning of discriminative features and mitigate overfitting, data augmentation strategies have been integrated into fine-grained image classification frameworks. Zhao et al. [7] advocated for the enrichment and augmentation of rock image data using Generative Adversarial Networks (GAN), aiming to enhance model performance by obtaining superior data samples.

1.1. Common Fine-Grained Image Classification Techniques

Due to the impact of the camera's location and angle and the resolution of the camera equipment, the objects to be classified in rock images typically have arbitrary locations and do not fill the entire image. The localization of classification targets can eliminate the interference of other elements in the background of the image. Most localization methods use bounding boxes to present the position of objects and mark the regions in the images that contain objects. Several localization methods determine the position of objects based on features generated by convolutional neural networks [8–11]. R-CNN approaches [10] involve the use of techniques such as selective search [12] to generate several candidate regions in the detection image and extract the corresponding feature vector from each region, with the most accurately classified region then identified as the target region. This approach requires manual annotation of the bounding boxes of the object parts, which consumes a significant amount of time and computational resources. Therefore, methods that use weakly supervised localization [13–16] have been introduced, such as RA-CNN [16], which continuously enlarges the area of interest in the convolutional layer and uses the enlarged image as input; the obtained classification result is then considered a part of the final result. S3N [17] determines the object region based on the maximal response location for each category in the image. While these methods achieve satisfactory results without the need for bounding box annotations, they require a two-stage process of localization and classification, and the presence of multiple inputs and networks can lead to complex models with lower efficiency.

In rock image samples, the classification of rock types largely depends on complex visual characteristics such as color and texture variations. High-quality features are particularly important in fine-grained image classification, as the visual differences between fine-grained images are often minimal. Building a robust feature representation has been extensively studied for fine-grained image classification. One of the earliest representative methods is Bilinear CNN [18], which aggregates features extracted by two CNNs to generate higher-order features representing more complex and higher-dimensional characteristics. Subsequent methods inspired by this approach include several methods [19–23] that rely on multiple feature extraction modules for multi-view image samples that are cropped, erased, masked, and zoomed for each module's input. These methods can extract richer feature representations and achieve better results, but cannot ascertain which features truly differentiate between classes or which are unique within a class. Rao [24] proposed a causal inference-based attention learning method that encourages the network to learn more effective visual attention. Inspired by this, fine-grained image classification methods based on contrastive learning have been proposed. Contrastive learning can obtain key features by comparing pair of images in order to distinguish them; this approach is especially effective for fine-grained classification, as it is able to detect subtle differences. CIN [20] uses the channel correlation between samples to bring positive pairs closer through channel interaction while pushing negative pairs apart, whereas PCA-Net [21] and API-Net [22] learn common and distinguishing features through channel interactions by simultaneously inputting positive/negative sample pairs. Fine-grained image classification methods based on contrastive learning have achieved the best results to date; however, the existing problem is that while the similar features of samples within the same class and different features of samples from different classes are used as bases for discrimination, these are not correct reasons for differentiation. For example, in a case where a pair of images belonging to

different classes share the same background, the model might improperly use this similarity as a basis for classification.

1.2. Lightweight CNNs Models for In Situ Rock Classification

Although this kind of method solves the problem of professionals with rich geological knowledge and experience being in very high demand, the process of rock identification and classification must still take place within the laboratory [25]. Moreover, the research results cannot be directly applied by geological surveyors working in the field [26], preventing effective real-time feedback. Wang et al. [27] designed a lightweight deep neural network with depthwise separable convolutions based on MobileNets and the transfer learning method, which they used to develop a lithology recognition model for rock images in the field. However, their model was not optimized or compressed in detail, and they did not compare it to other CNN-based models. The CNN backbone often refers to feature extraction layers and the generation of a representation vector for use in downstream tasks such as classification. Baraboshkin et al. [28] compared the effectiveness and efficiency of four types of mainstream CNN backbone models and discussed the importance of convenience and productivity in rock classification. However, their proposed method cannot be deployed on mobile devices. Fan et al. [29] proposed a rock classification model based on MobileNet and SqueezeNet architectures. They tested the model on mobile devices and presented detailed time and space consumption, proving that their model is applicable in the field. The above methods attempt to improve classification accuracy and efficiency by employing different CNN backbones, but do not consider two points: first, in fine-grained classification tasks it is important to preserve the rich features in rock images while applying a lightweight model, as multiple features can determine correctness; second, additional model optimizations such as compression and distillation need to be carried out in order to improve their potential and efficiency.

1.3. Improving Lightweight Deep Neural Networks by Knowledge Distillation

Although some fine-grained image classification algorithms have addressed the limitation of requiring additional annotated information and have achieved target extraction under unsupervised or weakly supervised conditions, balancing model accuracy and model size remains challenging. Pursuing high accuracy necessitates increased complexity and computational demands, leading to longer inference and prediction time costs. In the context of rock image classification, which often involves mobile environments, it is essential to address both size and accuracy. One of the current approaches for achieving this goal involves improving the classification accuracy of small models without expanding their size.

1.4. Refining the Decision Boundaries of Neural Networks

Using the trained model to predict all coordinate points in the sample space, the boundaries between different classes can be observed across all points in the sample space, ultimately yielding the model's decision boundaries. Activation boundaries refer different states of neurons in one layer between the deep network and formed regions. The shape of the decision boundary and the distance of sample points from the boundary are crucial in fine-grained classification tasks. Existing large-margin learning methods [30] mostly focus on fine-grained classification problems, such as face recognition and voiceprint recognition. The goal of large-margin learning algorithms is to increase the distance between sample points and the decision boundary. The advantages of the resulting large-margin classifiers include better generalization and robustness to input perturbations, which are highly effective for situations involving a large number of fairly similar samples and a few difficult samples, as is the case in fine-grained classification tasks. In many classical machine learning algorithms, such as Support Vector Machine, the distribution of sample points in the sample space is fixed and it is only necessary to solve for the optimal decision boundary. However, in deep learning models both the representation of samples and the

decision boundaries are learned through the backpropagation algorithm. For nonlinear deep networks, the distances between the sample points and the decision boundary are often replaced by approximate values, making optimization of the decision boundary in deep learning more complex and difficult to compute.

There are several methods for expanding the margin in the current deep learning literature. Sokolic et al. [31] aimed to limit the local variations of the model by minimizing the Lipschitz constant such that it does not exceed a certain constant by using L2 regularization during the model optimization process to prevent arbitrary expansion of model parameters, thereby enhancing the model's generalizability. Additionally, introducing a margin in the model's decision space (such as the probability space of model outputs) or representation space can be a more direct approach. S. Sun et al. [32] noted that using softmax combined with cross-entropy as the loss function does not inherently maximize the margin, and advocated for an additional penalty term to encourage a larger margin. In the field of face recognition, large margin losses based on softmax have been widely discussed, including in the context of L-Softmax [33] and a unified framework for softmax-based margin loss [34]. Furthermore, increasing the margin in the representation space is a common approach, with local linearization [35] as one representative method. By analyzing scenarios where sample perturbations lead to incorrect predictions, then combining the results with Taylor expansion, it is possible to approximate the specific form of the perturbations, which can serve as a representation of the margin.

Similar to large-margin learning, metric learning concentrates on learning a distance function in order to effectively measure similarities and differences between samples. Metric learning aims to learn a distance function tailored to the specific task. This distance function is designed such that similar samples are closer to each other and dissimilar samples are farther apart. In deep neural networks, metric learning measures the similarity of pairs of samples through a loss function, resulting in better learning of classification decision boundaries and improved model generalizability. Wang et al. [36] utilized a sum-based margin in the cosine space, setting the decision boundary within the cosine space so as to avoid the issue of inter-class samples being unconstrained by the angular distance. The center loss [37] adds a regularization term on top of the softmax loss, pushing apart different class samples in the feature space.

In neural networks, the activation state of the neurons is very important. Montu-far [38] proposed that neural networks represent complex functions with combinations of activation boundaries. Pan and Srikumar [39] demonstrated that the decision boundaries of neural networks are composed of combinations of these activation boundaries. Therefore, in knowledge distillation, transferring activation boundary information from teachers to students can convey more crucial classification information to the student models, especially in fine-grained classification problems. This is because models for fine-grained classification often have more complex decision boundaries, which is one of the bases for the methods adopted in this paper.

Figure 1 illustrates the misclassifications that arise from decision boundaries in general and fine-grained image classifications; the dots represent samples, the solid lines represent the ideal decision boundary of a small model, and the dashed lines represent the model's actual decision boundary. The different shades of background color indicate the model's activation response to a certain class; in (a), this refers to general image classification, while in (b) it refers to fine-grained image classification, with misclassified samples marked by red circles.

The key challenges in fine-grained image classification include: (1) considerable intra-class variance, as images of identical rock types may vary in background, perspective, and scale, leading to significant feature discrepancies within the same class; (2) minimal inter-class variance, characterized by highly similar samples among rock subclasses that share hues, textures, and shapes and are only distinguishable by nuanced differences; and (3) constrained recognition conditions in situ, as in field exploration, where limitations in image clarity and mobile device capabilities impede accuracy. Notably, small

models on mobile devices often fail to align with the decision boundaries of their larger counterparts, significantly diminishing the resulting fine-grained classification accuracy. Accurately modeling the decision boundaries of larger high-accuracy models remains a formidable challenge. Table 1 shows the advantages and disadvantages of different rock image classification approaches.

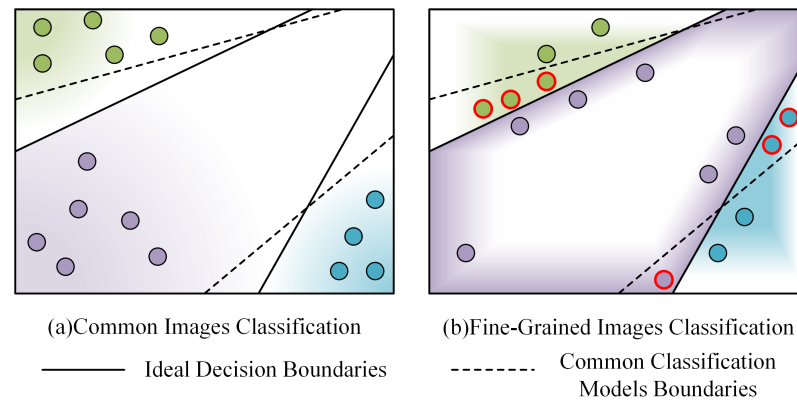


Figure 1. Instances that are easily confused in fine-grained image classification.

Table 1. Rock sample classification approaches.

Rock Classification Algorithm	Advantages	Disadvantages
SVM Classification	strong generalization ability	cannot deal with mass of rock image samples
Location-Classification CNN	excludes background noise	additional annotation
Contrast Learning-based Classification	captures detailed features	irrelevant comparison features lead to classification mistakes
FPCN (ours)	combines advantages of the above two models	larger time and space costs
Distilled Lightweight Model by FPCN (ours)		balances accuracy and consumption

This paper presents the design of a Feature Positioning Contrast Network (FPCN) model to effectively mine the critical features that dictate rock types. This approach distinguishes the essential features among rock classes and identifies unique characteristics of specific rock types. The model leverages these features to refine decision boundaries during the knowledge distillation process for in order to develop compact classification models, thereby boosting their performance. The proposed approach involves the following tasks: (a) Without relying on supplementary annotations for rock image location frames, the algorithm captures the fundamental position contours of the main rock body through feature maps produced by a pretrained model. (b) A self-supervised learning strategy is employed that hinges on contrastive learning within and between rock image classes in order to discern the pivotal features of rock types while concurrently filtering out non-essential comparative data. (c) By applying knowledge distillation within the FPCN model, the decision boundaries of the compact models are enhanced, resulting in improved classification accuracy. Furthermore, the paper includes tests and evaluations of compact rock classification models across various scales.

2. Knowledge Distillation-Based Lightweight Model Optimizing Framework

This chapter introduces the main framework for optimizing lightweight model methods, details the component modules of the FPCN, which contains rich fine-grained features as a teacher model, and describes the feature alignment and decision boundary transfer

processes involved in knowledge distillation. As shown in Figure 2, the proposed method primarily involves transferring the activation boundaries of the intermediate convolutional layers of the trained FPCN to the corresponding intermediate layers of the lightweight model through an amplification module. Cross-model correspondence between the shallow and deep layers is achieved through feature alignment and middle transform layers. The final training objectives of the lightweight model include similarity of the activation boundaries, predictive distribution of the FPCN, and the true labels.

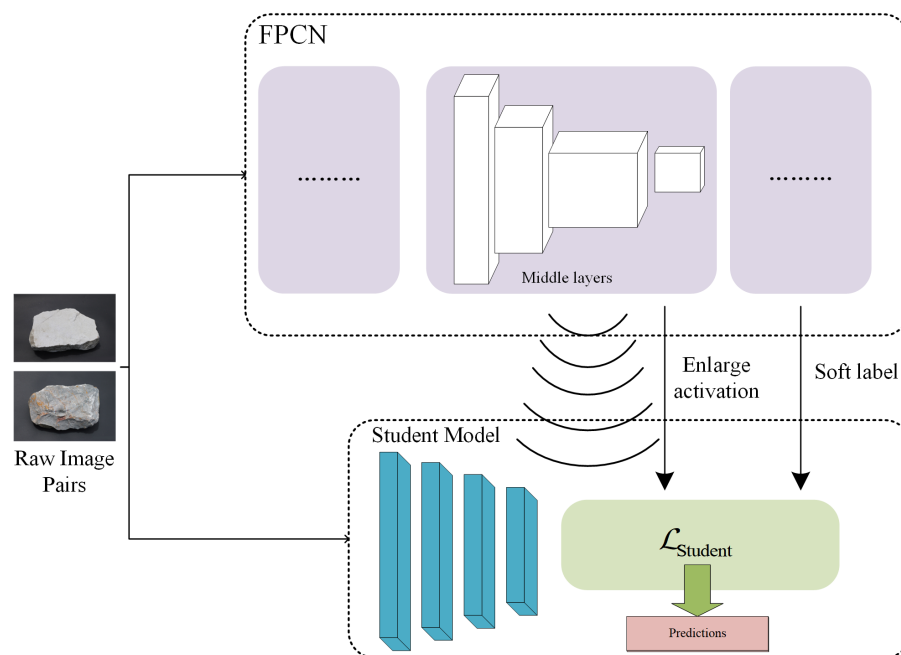


Figure 2. Framework for optimizing lightweight model method based on FPCN.

2.1. Feature Positioning Contrast Network

As depicted in Figure 3, the FPCN model consists of three main parts: Main Body Positioning, Part Contrast Learning, and Optimization Loss. The goal of the Main Body Positioning module is to identify the approximate area of the object within the image. Given the presence of interference from the background and other objects, which produce irrelevant contrast information, the positioning area aids the contrast module in focusing on the critical contrast locations. Our proposed method for detecting the maximal feature map area does not require additional bounding box annotations, instead obtaining the target position by generating highlighted connected regions in the feature map through a pretrained image detection network.

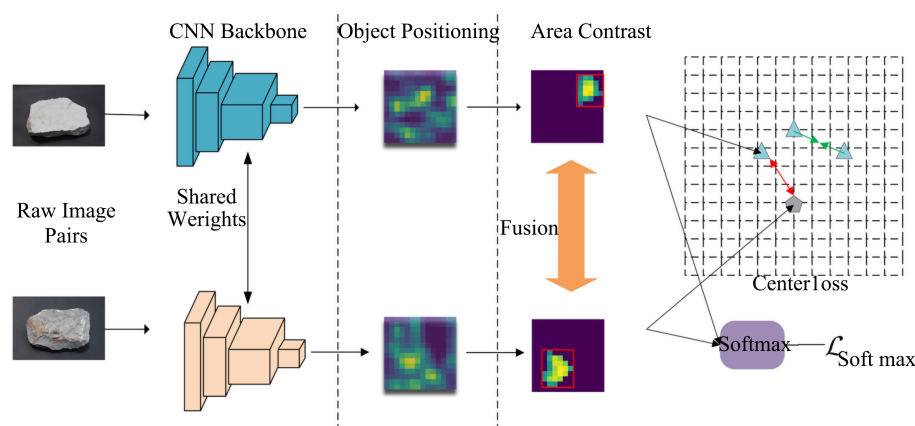


Figure 3. Feature Positioning Contrast Network (FPCN).

2.1.1. Feature Maps Area Positioning

Given an input image I , $F \in \mathbb{R}^{H \times W \times C}$ denotes the feature map obtained after the last convolutional layer of the backbone and H, W, C denote the height, width, and number of channels of the feature map, respectively. Due to interference caused by invalid contrast information from the background and other objects, the localized region can help to focus on the key contrast locations in the contrast module. Our proposed method for detecting the maximum region of the feature map does not require any additional labeling of the position frame; the target location is obtained by generating the highlighted connected regions of the feature map through a pretrained image detection network. The main steps are shown in Figure 4. To account for rock objects with different scales, the size of the maximum connected area is unique and maintains most of the area of the rock, helping to retain the rock features while reducing unrelated background features.

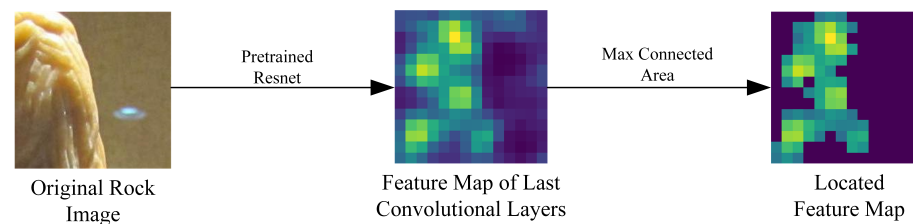


Figure 4. Steps in object positioning.

During experiments by Wei et al. [40], it was found that in an image of a single object the main region always corresponds to the position of the largest activation connected area in the feature map of the last convolutional layer. Using the VGG16 [41] model pretrained on the ImageNet dataset, the last layer's feature map greatly contributes to localization; however, using a single layer for localization in the ResNet model yields poor results with high-resolution image inputs. Due to the varying proportions of the main subject in different images, a single size of feature map cannot represent objects of various scales. Inspired by the Feature Pyramid Networks (FPN) proposed by Lin et al. [42], which have less semantic information in lower layers while providing accurate target locations and richer semantic information but coarser target locations in higher layers, in this paper we use two layers of feature maps to determine the main subject's position, as shown in Algorithm 1.

Algorithm 1: Generation of the largest connected region in feature maps

Input: Input $feature_b, feature_c$
 // $feature_b, feature_c$ denote feature map of conv5_bconv5_c

Output: Max Connect Area

```

1 while channel! = MaxChannel do
2    $sum\_b(x, y) = sum\_b(x, y) + feature\_b\_channel\_i(x, y);$ 
3    $sum\_c(x, y) = sum\_c(x, y) + feature\_c\_channel\_i(x, y);$ 
4  $mean\_b = sum\_b(x, y) / (x \times y);$ 
5  $mean\_c = sum\_c(x, y) / (x \times y);$ 
6 if  $feature\_b(x, y) > mean\_b$  then
7    $Mask\_b(x, y) = 1;$ 
8 else
9    $Mask\_b(x, y) = 0;$ 
10  $cand\_area\_b = GenArea(Mask\_b)$  // Connected region labeled as 1
11  $cand\_area\_c = GenArea(Mask\_c);$ 
12  $max\_area\_b = Max(List(cand\_area\_b));$ 
13  $max\_area\_c = Max(List(cand\_area\_c));$ 
14  $Area = max\_area\_b \cap max\_area\_c;$ 

```

2.1.2. Part Comparison Learning

In order for the model to learn common features within classes and differential features between classes, we propose a between-objects comparison learning module, as shown in Figure 5. In one training batch, there exist different image pairs of different classes or same class; for every pair in one batch, we first obtain the located feature map through object positioning and input it to $f(x)$, which presents a single linear layer, and output the fusion feature, which contains the similar and unique features for each image (marked in red and green, respectively). The fusion features point out the key feature for judging the rock types. Therefore, the element-wise multiplication between fusion features and scaled features generates modified feature that can activate both the unique and common features of the key features corresponding to the sample.

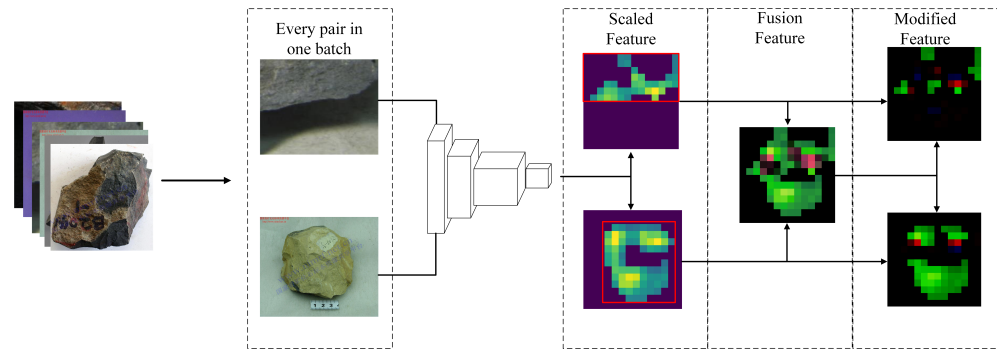


Figure 5. Process of part comparison learning.

This is shown in (1), where z_i is the feature representation of the i th sample, z_{i+} is its positive sample, z_{i-} is its negative sample, $\cdot$ is used to calculate the similarity between the samples, and τ is a temperature parameter that controls the shape of the loss function.

$$\mathcal{L}_c = -\log \frac{\exp(z_i \cdot z_{i+} / \tau)}{\sum_{i=0}^K \exp(z_i \cdot z_{i-} / \tau)} \quad (1)$$

The overall optimization process of the model during the training phase is shown in Equation (2), with $\mathcal{L}_{class}$ using the cross-entropy loss function between category predictions. The total loss is expressed as the sum of the classification loss and center loss, with λ being a hyperparameter that balances the two losses.

$$\mathcal{L} = \mathcal{L}_{class} + \lambda \mathcal{L}_c \quad (2)$$

Due to the limitation of batch size in the training process, for a certain sample in each batch the features are compared with other samples in the batch, then the model is trained by calculating the similarity with positive and negative examples. The process is shown in Algorithm 2.

Algorithm 2: Feature contrast learning

Input: Input z_i, z_{i+} , backbone f and g in FPCN

Output: modified model f, g

```

1 for batch  $\in \{1, \dots, \text{maxbatch}\}$  do
2   for  $i \in \{1, \dots, N\}, j \in \{1, \dots, N\}$  do
3      $s_{+,i,j} = z_i^\top z_{i+} / (\|z_i\| \|z_{i+}\|)$ 
4      $s_{-,i,j} = z_i^\top z_{i-} / (\|z_i\| \|z_{i-}\|)$ 
5   define  $\mathcal{L}(i, j)$  as  $\mathcal{L}(i, j) = -\log \frac{\exp(s_{i,j+} / \tau)}{\sum_{k=1}^N (\exp(s_{i,k+} / \tau) + \exp(s_{i,k-} / \tau))}$ 
6    $\mathcal{L} = \mathcal{L}_c$ 
7   update networks  $\{f, g\}$  to minimize  $\mathcal{L}$ 
8 return backbone network  $f(\cdot)$  and  $g(\cdot)$  in FPCN

```

2.2. Feature Alignment

Despite the height and width being consistent, it is necessary to add an intermediate layer transformation during the alignment process due to the inconsistent number of channels in the feature maps output by corresponding convolutional layers of different models. In this paper, the intermediate layer transformation refers to the connector structure used in Fitnets proposed by Romero et al. [43]. Assuming the same input image I , the feature map sizes obtained from the i th convolutional layer which need to be aligned in the teacher model and student model are $H \times W \times C_a$ and $H \times W \times C_b$, respectively. In order to change only the number of channels, a 1×1 convolution kernel is used in the corresponding convolutional layer to map the channel number C_s of the student model to the channel number C_t of the teacher model. The convolutional layer used in the i th layer is shown in Table 2, where C_{is} and C_{it} represent the number of channels of the feature maps output at the i th layer from the student and teacher models, respectively.

Table 2. Structure of the transformation layers.

Connect Layers	Input	Output	Kernel
Connector1	$H_1 \times W_1 \times C_{1s}$	$H_1 \times W_1 \times C_{1t}$	$1 \times 1, \text{ stride} = 1, \text{ padding} = 0$
Connector2	$H_1 \times W_1 \times C_{2s}$	$H_1 \times W_1 \times C_{2t}$	
Connector3	$H_1 \times W_1 \times C_{3s}$	$H_1 \times W_1 \times C_{3t}$	
Connector4	$H_1 \times W_1 \times C_{4s}$	$H_1 \times W_1 \times C_{4t}$	

In the experiments described in this chapter, the teacher and student models used the activation feature maps of four convolutional layers; thus, the total activation boundary amplification loss function should include the sum of the loss values of each layer. Because the width and height of the feature maps are halved after each block, leading to a reduction in the number of pixels calculated to a quarter of the previous layer, using the L2 norm results in the ratio of each layer's loss value being reduced to half that of the previous layer. Thus, to balance the feature weights of the four convolutional layers, the loss values of the lower layers are correspondingly reduced, as shown in Equation (3).

$$\mathcal{L}_1(\mathbf{I}) = (\mathcal{L}_{C1}/8) + (\mathcal{L}_{C2}/4) + (\mathcal{L}_{C3}/2) + \mathcal{L}_{C4} \quad (3)$$

2.3. Activation Boundary Amplification Module

Due to the importance of activation boundaries for fine-grained tasks, it is insufficient to simply use the neuron's activation values for distillation constraints. Instead, the neuron's activation regions should be used. To ensure that the lightweight model's decision boundaries more closely align with those of the location-contrast network, it is crucial to focus on whether the corresponding neuronal states are the same. The value of $if(x)$ indicates whether a neuron is activated, as shown in Equation (4).

$$if(x) = \begin{cases} 1, & \text{if } x = 0, \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

To make the decision boundaries of the FPCN and the small model more similar, Equation (5) describes the loss of the small model. The activation state of the neuron in the teacher model $if(T(\mathbf{I}))$ is equivalent to the category label. If the teacher's neuron is activated, then the response of the student's neuron should be greater than 0; conversely, if the teacher's neuron is not activated, then the response of the student's neuron should be less than 0, as follows:

$$\mathcal{L}(\mathbf{I}) = \| if(T(\mathbf{I})) \odot \sigma(\mu \mathbf{1} - S(\mathbf{I})) + (\mathbf{1} - if(T(\mathbf{I}))) \odot \sigma(\mu \mathbf{1} + S(\mathbf{I})) \|_2^2 \quad (5)$$

where $\odot$ is the element-wise multiplication between vectors, while $\mathbf{1}$ represents a vector of length equal to the output size of $T(\mathbf{I})$ with all values being 1. The alternative loss assigns

a squared penalty to neurons in the student model that have different activation states from the teacher, without concern for neurons with the same activation values. In our experiments conducted with this method, the layers from which activation values were extracted were all convolutional layers, and each layer's activation values corresponded to the values of the convolutional layer's feature maps. Thus, the above formula can be written as shown in Equation (6).

$$\mathcal{L}_1(\mathbf{I}) = \sum_i^H \sum_j^W \mathcal{L}(\mathbf{I}) \quad (6)$$

Because the training set does not include all data of rock images and new rock images are constantly being produced during exploration, the training goal becomes modeling the relationship between input and output on the existing training dataset. As the training dataset is a sample of the real data distribution, the optimal solution on the training dataset often deviates from the real optimal solution, although it should be noted that this discussion does not consider model capacity. In knowledge distillation, as we already have a model with good performance and strong generalization ability in the FPCN, we can allow the student model to learn the generalization ability of the FPCN directly when using the FPCN to distill the student model. An efficient method for transferring generalization ability is to use the probabilities of categories output by the softmax layer as soft targets.

In the output of the softmax layer, both the positive and negative examples contain a great deal of information; for instance, the probability associated with certain negative labels is much higher than that of others. In the traditional hard-target training process, all negative labels are treated uniformly, meaning that the training method with knowledge distillation provides more information per sample to the student model than the traditional training method.

Finally, the similarity of the activation boundaries, the soft labels for the classification of the teacher model, and the hard labels for the classification of the student model are combined as the comprehensive optimization objective, as shown in Equation (7).

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_{soft} + \mathcal{L}_{hard} \quad (7)$$

3. Experiments and Discussion

This chapter first explains the training environment used for training the FPCN and optimizing the lightweight models using the FGVC-ROCK dataset. Then, it compares the FPCN with existing fine-grained image classification models, lists the performance improvements of lightweight models, and discusses how each module contributes to the overall effectiveness.

FGVC-ROCK Dataset: The large-scale Fine-Grained Rock Image Dataset contains a variety of rock images with complex backgrounds along with multiple shapes and proportions (Table 3).

Table 3. FGVC-ROCK dataset.

Rock Type	Number
Dolomite	7924
Marble	11,658
Basalt	19,678
All	39,620

Training FPCN Experimental Setup: We used the PyTorch deep learning framework and performed training on a computer equipped with an RTX A5000 (24 GB) GPU and 42 GB of memory. FPCN uses a ResNet-101 pretrained on the FGVC-ROCK dataset as its backbone network. First, each input image was resized to 448×448 . The training process employed a stochastic gradient descent optimization algorithm (SGD) with a momentum

of 0.9, and the training was set for 100 epochs (actually converging around 80–90 epochs) with a batch size of 42 (14 images per class) and an initial learning rate of 0.01. When constructing pairs of input images, mask feature maps and contrast vectors were generated for any two samples in each batch, and the model was optimized as a whole.

Distillation Experimental Setup: The experimental environment in this chapter was the same as when training FPCN. The teacher model used the FPCN network trained on the FGVC-ROCK dataset, while the student models included models suitable for deployment on mobile devices, including ResNet18, ResNet34, MobileNet V1, MobileNet V2, ShuffleNet V1, and ShuffleNet V2. The backbone networks of the student models were also pretrained on the ImageNet dataset. The entire distillation model was trained for 100 epochs, with the first 60 epochs used for distillation training, setting an initial learning rate of 0.01, which was reduced to one-tenth every 20 epochs. The setup for the Image Transform is shown in Table 4.

Table 4. Transforms used in training and testing.

Operation	Train Input Size	Output Size	Operation	Test Input Size	Output Size
resize	Original Image Size	(512,512,3)	resize	Original Image Size	(512,512,3)
Random Cropping	(512,512,3)	(448,448,3)	Center Cropping	(512,512,3)	(448,448,3)
Random horizontal Flipping	(448,448,3)	(448,448,3)			
Normalization	(448,448,3)	(448,448,3)	Normalization	(448,448,3)	(448,448,3)

Convolutional Layer in Different CNN Backbones: To uniformly denote the corresponding convolutional layers of different models, the labels C_1, C_2, C_3, C_4 are used to represent the four aligned convolutional layers. The input image I has a resolution of $448 \times 448 \times 3$. After passing through the first two convolutional layer blocks of the backbone network, a feature map of $\frac{1}{4}$ resolution, denoted as $conv_2(112 \times 112)$, is obtained. Subsequently, by passing through the remaining convolutional layers, feature maps of different scales are obtained, namely, $conv_3(56 \times 56)$, $conv_4(28 \times 28)$, and $conv_5(14 \times 14)$. The sizes of these feature maps correspond to different convolutional layers in the teacher and student models. By using the similarity of the corresponding feature maps, the student model learns the activation boundaries of the teacher model, thereby learning the decision boundaries of the teacher model. The correspondences between the output sizes of each convolutional layer in the teacher and student models are shown in Table 5.

Table 5. Connection of convolutional layer.

Model	Block1	Block2	Block3	Block4
ResNet-101 (teacher)	conv_2	conv_3	conv_4	conv_5
ResNet-18	conv_2	conv_3	conv_4	conv_5
ResNet-34	conv_2	conv_3	conv_4	conv_5
MobileNetV1	Conv dw/s2(1)	Conv dw/s2(2)	Conv dw/s2(3)	Conv dw/s2(4)
MobileNetV2	Conv dw/s2(1)	Conv dw/s2(2)	Conv dw/s2(3)	Conv dw/s2(4)
ShuffleNetV1	Conv1 MaxPool	Stage2	Stage3	Stage4
ShuffleNetV2	Conv1 MaxPool	Stage2	Stage3	Stage4
Output Size	112×112	56×56	28×28	14×14

3.1. FPCN Model Effect

As shown in Table 6, we compared the method proposed in this article with basic image classification models and fine-grained classification models. Using the same rock image dataset, the proposed method achieved the best accuracy, reaching 93.1%. Compared with the next best models, PCA-Net and API-Net, both of which are based on contrastive learning, our method adds an unsupervised main body localization module, which achieves

the filtering of invalid contrastive information, resulting in better effects. In comparison with MMAL-Net’s simultaneous localization of the main body and multi-part multi-view learning method, our approach considers the proportion of the object in the image and applies a soft mask based on the size of the ratio, supplementing the feature information when the target object’s localization is imprecise or the target object is small.

Table 6. Comparison of different fine-grained classification models.

Models	Training Accuracy	Test Accuracy	Backbone
Bilinear CNN (2015)	85.4	85.3	VGG16
CIN (2020)	93.2	89.8	ResNet-101
PCA-Net (2021)	94.9	90.8	ResNet-101
API-Net (2020)	99.8	91.2	ResNet-101
MMAL-Net (2021)	99.7	91.9	ResNet-50
FPCN	99.8	93.1	ResNet-101

The impact of the number of image pairs in a single batch on the results can be seen in Figure 6; increasing the number of image pairs significantly improves the accuracy of all three models, indicating that the models learn more contrastive feature information and can more accurately categorize images that are difficult to distinguish. In the process of increasing contrast samples, the method proposed in this paper surpasses the other two contrast learning-based classification models. The possible reasons are as follows. First, when there are fewer contrast samples, the model receives limited feature information and the feature maps processed by the mask contain insufficient information, limiting the number of features extracted by the network. Second, when there are enough contrast samples, the model extracts a large number of contrast features, including some incorrect contrast features (such as the same background in the same class of images). Our proposed model can filter out more critical information from a large number of contrast features through object localization, thereby achieving better results.

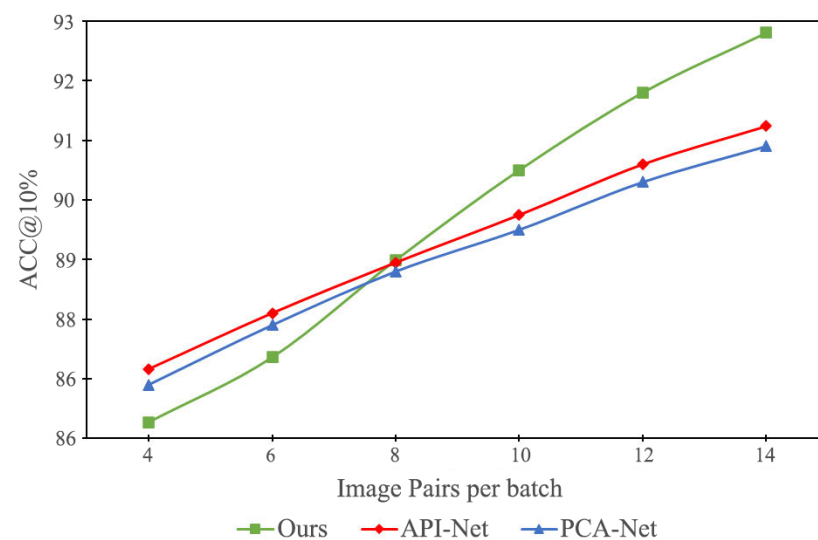


Figure 6. Effect of the number of sample pairs in a single batch for different models.

3.2. Lightweight Distillation Effect

The results when training smaller student models using the model proposed in this paper as the teacher model are shown in Table 7. Each column in the figure represents various parameters of different models, including the total number of model parameters; the number of floating-point operations (FLOPs), which presents the floating point operations per second, shows the device’s requirements in terms of computing capacity. The inference time for a single sample presents the time between inputting the picture and

outputting the classification result. The test set accuracy on the rock image dataset uses the knowledge distillation method by Hinton et al. and the knowledge distillation approach focusing on decision boundaries. The accuracy results of the small models distilled from the FPCN model are higher than those of the baseline models, achieving a balance between model efficiency and effectiveness to some extent.

Table 7. Distillation improvements on lightweight models.

Model	(Teacher) FPCN	ResNet-18	ResNet-34	ResNet-50	MobileNet V1	MobileNet V2	ShuffleNet V2 1.5	ShuffleNet V2 0.5
Params (Million)	46.06 M	11.89 M	22.00 M	25.76 M	4.2 M	3.4 M	2.4 M	0.3 M
FLOPS	28,210 M	7294 M	14,713 M	16,529 M	569 M	326 M	303 M	43 M
Inference Time (s)	13.98	2.61	5.63	7.50	1.13	0.78	0.68	0.16
Baseline Accuracy (%)	93.1	71.8	74.7	76.5	74.5	75.5	71.4	70.3
Distilled Accuracy (%)		72.7 (+0.9)	76.1 (+1.4)	77.1 (+0.6)	76.3 (+1.8)	77.4 (+1.9)	72.2 (+0.8)	71.0 (+0.7)

The ResNet series (ResNet-18, ResNet-34, ResNet-50) and the MobileNet and ShuffleNet versions all achieve improved accuracy through distillation, which proves that the knowledge distillation technique can effectively transfer knowledge from the teacher model to the smaller student model while improving the latter's performance. For the ResNet series, the improvement in accuracy due to distillation ranges from 0.6% to 1.4%. This shows that knowledge distillation is effective in improving performance even for relatively large models. In smaller models (e.g., MobileNet V1 and V2, ShuffleNetV2 1.5 and 0.5) the accuracy improvement is even more significant, ranging from 0.7% to 1.9%. This emphasizes the potential of knowledge distillation in compressing models and reducing computational resource requirements, especially for applications on mobile or edge devices. The inference time data shows that smaller models are more accurate and also run faster, which is important for applications that require fast response times. Among the ResNet models, the accuracy of ResNet18, ResNet34, and ResNet50 is 72.7%, 76.1%, and 77.1%, respectively, 0.9%, 1.4%, and 0.6% higher than the models without knowledge distillation (i.e., the models trained only with $\mathcal{L}_{hard}$) and 0.6% higher than the proposed models; the largest ResNet50 model has reductions of nearly 50% in the number of parameters, number of floating-point operations, and inference time compared to the proposed FPCN model, which are 25.76 M, 16,529 M, and 7.5 s, respectively. Among the various models we tested, MobileNetV2 has the highest accuracy improvement of 1.9%, and the distillation models of MobileNetV1 and MobileNetV2 have the best overall performance, which is only one-tenth of the FPCN in terms of various resource consumption indexes. Moreover, the accuracy is 5% higher than that of the two ShuffleNet networks. For ShuffleNetV2 1.5 and ShuffleNetV2 0.5, although the resource consumption is very small and the number of parameters is only 2.4 M and 0.3 M, the model accuracy is lower and the distillation effect is not significant, which may be due to the fact that the model with the number of parameters may not be able to fit more complex decision boundaries, meaning that the proposed optimization of activation boundaries leads to greater variation of the decision boundaries and leading to part of the originally correct samples being misclassified.

To further demonstrate the effectiveness of closer decision boundaries, the relationship between the similarity of neuron activation in the last layer of each convolutional layer of ResNet and the final classification accuracy is listed in Table 8. It can be seen from the table that there is a positive overall correlation between neuron activation similarity and accuracy. For the ResNet-18 and ResNet-34 models, the neuron activation similarities for the shallowest and deepest layers reach 95.7%, 98.6%, 72.7%, and 76.1%, respectively, each showing nearly 1% improvement compared to the knowledge distillation methods without optimized activation boundaries.

Table 8. Neuron similarity in lightweight models.

Models	Similarity(%)				Accuracy (%)
	conv_2	conv_3	conv_4	conv_5	
KD ResNet-18	93.3	82.6	70.3	59.8	71.8
KD+DB ResNet-18	95.7	93.9	89.0	76.5	72.7
KD ResNet-34	96.7	96.5	93.9	83.3	74.7
KD+DB ResNet-34	98.6	98.0	95.1	85.6	76.1
KD MobileNetV2	98.1	86.9	73.9	73.4	75.5
KD+DB MobileNetV2	99.1	98.9	95.7	82.2	77.4
KD ShuffleNetV2 1.5	92.4	81.8	69.6	59.2	71.4
KD+DB ShuffleNetV2 1.5	94.7	93.0	88.1	75.7	72.2

In all models, the method of knowledge distillation with decision boundary (KD+DB) significantly improved the neuron similarity across all convolutional layers (*conv*₂, *conv*₃, *conv*₄, *conv*₅) compared to using knowledge distillation alone. This indicates that the KD+DB method can more effectively transfer the feature representation ability of the teacher model to the student model, more closely mimicking the teacher model's behavior at deeper levels.

In terms of accuracy, all models using the KD+DB method outperformed those using only the KD method. This shows that by better simulating the neuronal behavior of the teacher model, the student model can learn the decision boundaries of the teacher model more accurately, resulting in improved overall performance of the model.

In comparing the models with different structures, as the model depth of the ResNet series increases (from ResNet-18 to ResNet-34), both the neuron similarity and accuracy improve using either the KD or KD+DB method. This suggests that deeper models are better able to learn and simulate the characteristics of the teacher model during knowledge distillation. For MobileNetV2 and ShuffleNetV2, the improvement in neuron similarity is particularly significant compared to the ResNet series when using the KD+DB method, which is especially the case in deeper convolutional layers. This indicates that the KD+DB method may enhance model performance more effectively for lightweight models with higher computational efficiency.

The temperature used during knowledge distillation for fine-grained tasks also impacts the outcomes. The temperature level alters the degree of attention the student model pays to negative labels during the training process; at lower temperatures, less attention is paid to negative labels, especially those significantly below the average, while at higher temperatures the values associated with negative labels are relatively higher, leading the student model to pay more attention to negative labels. Indeed, negative labels can contain certain types of information, particularly those with values significantly above the average; however, due to the training process used by the FPCN, the negative label part contains more noise, with lower values of the negative label indicating less reliable information. Therefore, the choice of temperature has a significant impact on the distillation outcomes. For fine-grained classification tasks, the impact of negative labels is mainly divided into two categories. In the first category, the negative label scores assigned by the FPCN model are very close to the positive label scores. This indicates ne of two scenarios; first, the existence of similar samples between different classes, providing crucial detail information for fine-grained classification decisions, in which case paying more attention to negative label information aids training; or second, misclassification by the FPCN teacher model, resulting in high negative label scores, in which case overemphasis leads to incorrect training information. In the second category, the negative label scores assigned by the FPCN model have a large gap compared to the positive label scores, in which case the impact of negative labels is smaller. For scenarios in the first category, choosing different temperatures can enhance the effectiveness of knowledge distillation in fine-grained sample contexts. Figure 7 shows the impact of knowledge distillation at different temperatures on model performance.

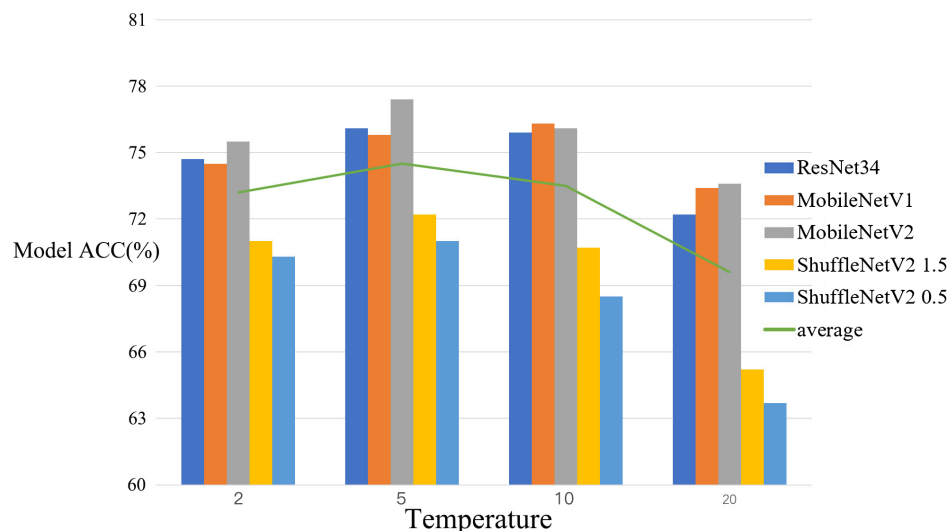


Figure 7. Effect of different distillation temperatures on the accuracy of different models.

4. Conclusions

In this paper, a fine-grained image classification model based on deep learning is proposed to solve the problem of low classification accuracy on the part of existing methods for fine-grained rock image classification by mobile lightweight models. The proposed model is used to develop an improved lightweight model for mobile applications; specifically, a Feature-map Positioning and Contrastive Learning Network (FPCN) is proposed for weakly supervised fine-grained rock image recognition, guiding the generation of lightweight classification models on mobile devices by shaping the transfer decision boundaries. Our findings indicate that focusing on knowledge distillation around the decision boundaries enhances the accuracy of lightweight models.

Author Contributions: Conceptualization, M.M. and Z.G. (Zhiming Gui); methodology, M.M.; software, M.M.; validation, M.M. and Z.G. (Zhiming Gui); formal analysis, M.M.; investigation, Z.G. (Zhiming Gui); resources, Z.G. (Zhenji Gao); data curation, Z.G. (Zhenji Gao) and M.M.; writing—original draft preparation, M.M.; writing—review and editing, Z.G. (Zhiming Gui); visualization, M.M.; supervision, Z.G. (Zhenji Gao) and B.W.; project administration, Z.G. (Zhenji Gao) and B.W.; funding acquisition, Z.G. (Zhenji Gao). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China National Geological survey project Construction of Big Data Intelligent Prediction System for Mineral Resources under grant number DD20243267.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available under Feature Positioning Contrast Network at <https://doi.org/10.5281/zenodo.10099204> (accessed on 15 June 2024).

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

References

1. Stille, H.; Palmström, A. Classification as a tool in rock engineering. *Tunn. Undergr. Space Technol.* **2003**, *18*, 331–345. [CrossRef]
2. Liang, Y.; Cui, Q.; Luo, X.; Xie, Z. Research on classification of fine-grained rock images based on deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 5779740. [CrossRef] [PubMed]
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
4. Chatterjee, S. Vision-based rock-type classification of limestone using multi-class support vector machine. *Appl. Intell.* **2013**, *39*, 14–27. [CrossRef]

5. Guojian, C.; Peisong, L. Rock thin-section image classification based on residual neural network. In Proceedings of the 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 9–11 April 2021; pp. 521–524.
6. Pascual, A.D.P.; Shu, L.; Szoke-Sieswerda, J.; McIsaac, K.; Osinski, G. Towards natural scene rock image classification with convolutional neural networks. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–4.
7. Zhao, G.; Cai, Z.; Wang, X.; Dang, X. GAN Data Augmentation Methods in Rock Classification. *Appl. Sci.* **2023**, *13*, 5316. [\[CrossRef\]](#)
8. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
9. Yan, J.; Lin, S.; Bing Kang, S.; Tang, X. Learning the change for automatic image cropping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 971–978.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
11. Lin, D.; Shen, X.; Lu, C.; Jia, J. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1666–1674.
12. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [\[CrossRef\]](#)
13. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-based R-CNNs for fine-grained category detection. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Proceedings, Part I 13; Springer: Cham, Switzerland, 2014; pp. 834–849.
14. Zhang, H.; Xu, T.; Elhoseiny, M.; Huang, X.; Zhang, S.; Elgammal, A.; Metaxas, D. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1143–1152.
15. Huang, S.; Xu, Z.; Tao, D.; Zhang, Y. Part-stacked CNN for fine-grained visual categorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1173–1182.
16. Fu, J.; Zheng, H.; Mei, T. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4438–4446.
17. Ding, Y.; Zhou, Y.; Zhu, Y.; Ye, Q.; Jiao, J. Selective sparse sampling for fine-grained image recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6599–6608.
18. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear CNN models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1449–1457.
19. Ji, R.; Li, J.; Zhang, L. Siamese self-supervised learning for fine-grained visual classification. *Comput. Vis. Image Underst.* **2023**, *229*, 103658. [\[CrossRef\]](#)
20. Gao, Y.; Han, X.; Wang, X.; Huang, W.; Scott, M. Channel interaction networks for fine-grained image categorization. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10818–10825.
21. Zhang, T.; Chang, D.; Ma, Z.; Guo, J. Progressive co-attention network for fine-grained visual classification. In Proceedings of the 2021 International Conference on Visual Communications and Image Processing (VCIP), Munich, Germany, 5–8 December 2021; pp. 1–5.
22. Zhuang, P.; Wang, Y.; Qiao, Y. Learning attentive pairwise interaction for fine-grained classification. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13130–13137.
23. Chen, J.; Li, H.; Liang, J.; Su, X.; Zhai, Z.; Chai, X. Attention-based cropping and erasing learning with coarse-to-fine refinement for fine-grained visual classification. *Neurocomputing* **2022**, *501*, 359–369. [\[CrossRef\]](#)
24. Rao, Y.; Chen, G.; Lu, J.; Zhou, J. Counterfactual attention learning for fine-grained visual categorization and re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 1025–1034.
25. Shu, L.; McIsaac, K.; Osinski, G.R.; Francis, R. Unsupervised feature learning for autonomous rock image classification. *Comput. Geosci.* **2017**, *106*, 10–17. [\[CrossRef\]](#)
26. Li, N.; Hao, H.; Gu, Q.; Wang, D.; Hu, X. A transfer learning method for automatic identification of sandstone microscopic images. *Comput. Geosci.* **2017**, *103*, 111–121. [\[CrossRef\]](#)
27. Wang, C.; Li, Y.; Fan, G.; Chen, F.; Wang, W. Quick Recognition of Rock Images for Mobile Applications. *J. Eng. Sci. Technol. Rev.* **2018**, *11*, 111–117. [\[CrossRef\]](#)
28. Baraboshkin, E.E.; Ismailova, L.S.; Orlov, D.M.; Zhukovskaya, E.A.; Kalmykov, G.A.; Khotylev, O.V.; Baraboshkin, E.Y.; Koroteev, D.A. Deep convolutions for in-depth automated rock typing. *Comput. Geosci.* **2020**, *135*, 104330. [\[CrossRef\]](#)
29. Fan, G.; Chen, F.; Chen, D.; Dong, Y. Recognizing multiple types of rocks quickly and accurately based on lightweight CNNs model. *IEEE Access* **2020**, *8*, 55269–55278. [\[CrossRef\]](#)

30. Guo, Y.; Zhang, C. Recent advances in large margin learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7167–7174. [[CrossRef](#)] [[PubMed](#)]
31. Sokolić, J.; Giryes, R.; Sapiro, G.; Rodrigues, M.R. Robust large margin deep neural networks. *IEEE Trans. Signal Process.* **2017**, *65*, 4265–4280. [[CrossRef](#)]
32. Sun, S.; Chen, W.; Wang, L.; Liu, X.; Liu, T.Y. On the depth of deep neural networks: A theoretical view. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
33. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. *arXiv* **2016**, arXiv:1612.02295.
34. Kobayashi, T. Large Margin In Softmax Cross-Entropy Loss. In Proceedings of the British Machine Vision Conference (BMVC), Cardiff, UK, 9–12 September 2019; Volume 2, p. 3.
35. Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; Bengio, S. Large margin deep networks for classification. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
36. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
37. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part VII 14; Springer: Cham, Switzerland, 2016; pp. 499–515.
38. Montufar, G.F.; Pascanu, R.; Cho, K.; Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
39. Pan, X.; Srikumar, V. Expressiveness of rectifier networks. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 2427–2435.
40. Wei, X.S.; Luo, J.H.; Wu, J.; Zhou, Z.H. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Trans. Image Process.* **2017**, *26*, 2868–2881. [[CrossRef](#)] [[PubMed](#)]
41. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
42. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
43. Adriana, R.; Nicolas, B.; Ebrahimi, K.S.; Antoine, C.; Carlo, G.; Yoshua, B. Fitnets: Hints for thin deep nets. *Proc. ICLR* **2015**, *2*, 1.

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.