A Novel ResNet50-based Attention Mechanism For Image Classification

Jingsi Zhang, Xiaosheng Yu, Xiaoliang Lei, and Chengdong Wu*

Faculty of Robot Science and Engineering, Northeastern University Shenyang 110819, China

Corresponding author. E-mail: wuchengdongnu@163.com

Received: Oct. 09, 2023; Accepted: Oct. 30, 2023

Image classification tasks often compress the neural network model to reduce the number of parameters, which leads to a decrease in classification accuracy. Therefore, we propose a novel ResNet50-based attention mechanism for image classification. ResNet50 network is used to extract image features and input the features into the graph neural network as node features. Then, packet convolution and depth-separable convolution are used to compress the residual network. The attention mechanism is introduced into the network backbone to make it focus on the important part of the neighborhood and help the branch network to extract key information. The accuracy of 5-way 1-shot task classification on three publicly available datasets reaches 86.32%, 92.21% and 92.19%, respectively. The proposed method has achieved remarkable results in image classification tasks.

Keywords: Image classification; ResNet50; attention mechanism; depth-separable convolution; packet convolution

http://dx.doi.org/10.6180/jase.202408_27(8).0004

1. Introduction

Image classification is an important part of computer vision research. With the complexity of classification tasks, traditional classification algorithms cannot meet the current needs [1, 2]. To solve this problem, deep convolutional neural networks are proposed. Subsequently, researchers found that the deeper the network, the better the performance, such as VGG (visual geometry group) network and ResNet (residual network) [3]. Among them, the residual network deepens the network depth by stacking the residual structure, and improves the effect of the model in image classification task. Later, a larger residual network is proposed. Although the performance is improved to a certain extent, the computation and storage capacity are significantly increased, which is difficult to meet the requirements of high real-time mobile devices and resource-limited embedded devices.

In order to reduce the deployment difficulty and alleviate the performance bottleneck of edge device deployment, a series of model compression schemes are proposed, which includes parametric quantization [4], low-rank decomposition [5], lightweight network design [6], and knowledge distillation [7]. The methods of lightweight network design and knowledge distillation have attracted much attention. Although this method can be applied to large-scale neural networks VGG and ResNet to achieve lightweight networks, the performance of neural networks will be affected to a certain extent when they are compressed. Subsequently, Xu and Liu [8] proposed a self-distillation method, using the neural network itself to construct a deep classifier as a teacher model. Then, according to the depth of the deep classifier, it was divided into several regions, adding modules in different regions, constructing shallow classifiers as student models, and finally conducting self-distillation training for all classifiers. At present, there are also improvement schemes to introduce attention mechanism into shallow classifiers, which make shallow classifiers more lightweight and further improve the distillation framework. However, when the number of deep classifiers is large, the calculation of the whole process is still large, the distillation efficiency is low, and the shallow classifiers still have room for lightweight.

In order to solve the above problems, a novel ResNet50-based attention mechanism for image classification is proposed. The network lightweight and lightweight self-distillation algorithm are combined to further lightweight the self-distillation frame by reducing the number of parameters of the shallow classifier, reducing the calculation amount and shortening the training time of self-distillation without affecting the distillation effect. The attention mechanism is introduced into the network backbone to make it focus on the important part of the neighborhood and help the branch network to extract key information. Through network lightweight, the large neural network can be lightweight to achieve model compression, and the compressed network can be self-distilled to ensure the classification accuracy.

2. Related works

The lightweight attention module [9] achieves the effect of enhancing the original features by generating an attention mask and performing dot product operations with the input feature map. Let the input feature maps of the module height H, width W, and number of channels Z be divided into n groups according to the number of channels, and n intermediate feature maps of height H, width W, and number of channels Z/n are obtained.

The global average pooling operation is performed on each group of intermediate feature graphs to obtain the feature graph g with height 1 , width 1 and channel number Z/n. Then, the initial attention mask is obtained by site multiplication between g and the group of intermediate feature maps, and its mean and standard deviation are calculated. Then, the feature map with height H, width W and channel number 1 is obtained by normalization. Then, the final attention mask is obtained by activation of sigmoid function, and site multiplication is performed with the group of intermediate feature maps. The output feature map of the group is obtained, the height is H, the width is W, and the number of channels is Z/n. Finally, the output feature maps of all groups are spliced into the final output feature map, the size of which is the same as the input feature map, that is, the height is H, the width is W, and the number of channels is Z. Since the whole attention module is mainly composed of global average pooling layer, the correlation of global and local features is used to generate attention mask, so the parameter number and calculation amount of this module can be basically ignored.

The main branch of ResNet50 residual structure contains 3 convolution layers, the middle layer is a convolution layer with convolution kernel size of 3 , and the front and rear layers are a convolution layer with convolution kernel size of 1 , which respectively play the role of feature graph reduction and dimension increase [10, 11]. The path branch is mainly composed of a convolution layer with convolution kernel size 1 , which is used to ensure that the feature map dimension of the branch is consistent with that of the main branch. Based on this, a lightweight ResNet50 residual structure is proposed, as shown in Fig. 1, to replace all ResNet50 residual structures and achieve lightweight ResNet50 networks. Fig. 1(a) shows an ordinary residual structure, and Fig. 1(b) shows an ascending residual structure that requires a shortcut branch consisting of an ordinary convolution layer with a convolution kernel size of 1 . The main branches of the two residual structures are composed of two point-by-point convolution layers and one deep convolution layer.

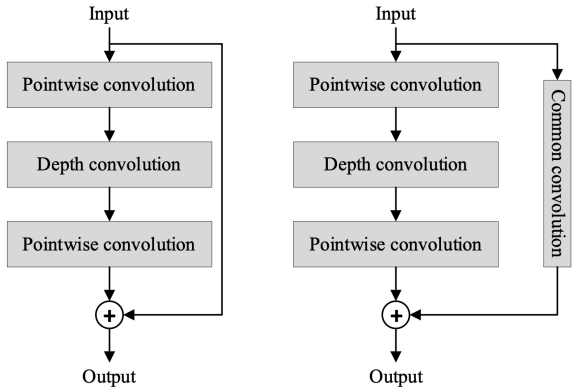

Fig. 1. Lightweight ResNet50 structure

3. Proposed image classification

This paper proposes a ResNet50 image classification algorithm based on attention mechanism. The model framework is shown in Fig. 2. Each circle with a different color in the figure represents the features of different samples. After ResNet50 processing, these feature vectors have changed. The model in this paper consists of two parts: the feature extraction Module based on ResNet50 network and the Dual Metric Module. The high-dimensional features extracted by the feature extraction module of ResNet50 network are used as the node features in the graph neural network. The double metric module uses cosine metric and Euclidean distance as two similarity measures to calculate the similarity between nodes as edge features in graph neural networks. After updating node feature and edge feature alternately, the category probability of the queried image is output at last.

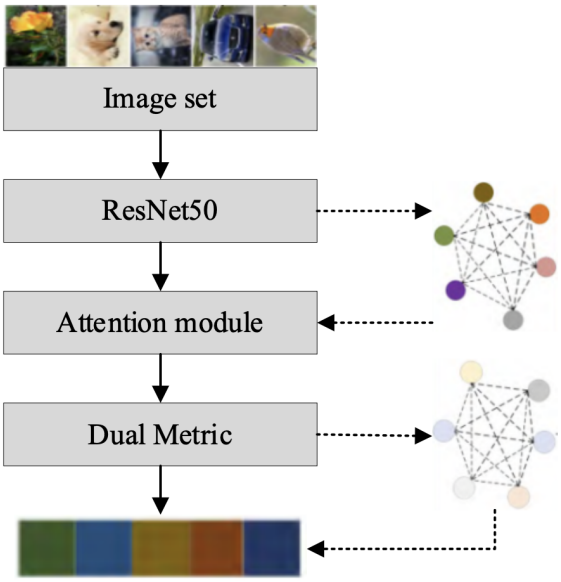

Fig. 2. Overview structure of proposed network

The shallow layer module is mainly composed of N-group convolution layer and an adaptive average pooling layer. Each set of convolution layers consists of two sets of depth-separable convolution layers and a lightweight attention module. In the first deep convolution process of a single set of convolution layers, the steps are set to 2 , the steps of the remaining convolution operations are set to 1 , and the number of groups N is determined by the position of the shallow classifier.

There are three shallow classifiers in the self-distillation frame. The number of convolutional layer groups of the shallow classifier is denoted as N_i, then the number of convolutional layer groups of the shallow classifier i is $N_i = 4 - i$, where $i = 1, 2, 3$. Suppose there are R samples $X = \{x_i\}_{i=1}^{R}$, and the samples have P categories, and the true labels corresponding to the categories are denoted $Y = \{y_i\}_{i=1}^{P}$. It is also assumed that there are C classifiers in the self-distillation frame, denoted $V = \{\theta_i\}_{i=1}^{R}$. softmax is added to the end of each classifier. At the same time, the temperature coefficient is introduced in the softmax, and the output label can be softened by modifying the value of the temperature coefficient. softmax after the temperature coefficient is introduced

$$a_i^c = \frac{\exp\left(z_i^c / T\right)}{\sum_j^c \exp\left(z_i^c / T\right)} \tag{1}$$

Where z_i^c is the output result of class i after the fully connected layer of classifier θ_c. a_i^c is the output probability of class i of classifier θ_c. When T is set to 1 , Eq. (1) is converted to normal softmax. The larger the T, the softer the label will be.

In this paper, ResNet50 is used as the feature extraction module of image classification, and the output image features are used as the node features of the graph neural network. It consists of convolution layer, linear layer, Patch Merging, Block block, global adaptive pooling layer and fully connected layer. First, the image is divided into 4×4 non-overlapping image blocks after a convolution operation. The image blocks are then converted into sequences through a linear embedding layer. In each Block, a self-attention mechanism is used to extract the features of the image. Subsequently, by sub-sampling Patch Merging, the width and height of the feature map are reduced, and the number of channels is increased. The deep features of the image are extracted through multiple Block and Patch Merging operations [12, 13]. Finally, these features are mapped to the final feature space through the fully connected layer. Using ResNet50 as the image feature extractor, we can obtain better global features of the image and improve the node feature representation of the graph neural network.

The Block contains four key modules: Layer Normalization (LN), Windowed Multi-head Self-Attention (W-MSA), Shifted Window Multi-head Self-Attention (SW-MSA), and shifted window multi-Head self-attention (SW-MSA) Multi-Layer Perceptron (MLP). The LN module is used to normalize input features to ensure that features of different channels have a similar distribution. The W-MSA module is used for multi-head self-attention computation in the window, and realizes the modeling and integration of the features in the window by paying attention to the fully connected feed-forward network, through multiple fully connected layers and activated nonlinear transformations to capture more successive actions. The Block block calculation process is as follows:

$$\hat{z}^l = W - MSA\left(LN\left(z^{l-1}\right)\right) + z^{l-1} \tag{2}$$

$$z^l = MLP\left(LN\left(\hat{z}^l\right)\right) + \hat{z}^l \tag{3}$$

$$\hat{z}^{l+1} = SW - MSA\left(LN\left(z^l\right)\right) + z^l \tag{4}$$

$$z^{l+1} = \mathrm{MLP}\left(LN\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1} \tag{5}$$

In Eqs. (2) to (5), $\hat{z}$ represents the output of the attention of many heads. z represents the output of the multi-layer perceptron.

In this paper, a multi-scale feature extraction network with attention mechanism (MFA) is proposed. Firstly, MFA

uses three one-dimensional convolution layers and maximum pooling layer to realize multi-scale strategy of receptive field. In addition, in order to realize the grouping multi-scale strategy, the original image vectors and feature vectors under different receptive fields are grouped respectively, and the number of groups is set by decreasing method. The larger the receptive field is, the smaller the number of groups is. At different scales, an LSTM is used for further image feature extraction. Finally, the features at different scales are added together to get the multi-scale features. In order to give full play to the multi-scale grouping strategy, MAF introduces an attention mechanism network at the output of LSTM at different scales to alleviate the phenomenon of gradient disappearance.

MAF mainly uses two fully connected layers to process the hidden layer state vectors of features of different levels in LSTM, then generates weights, and calculates the weighted and summed information u. The specific formula is as follows:

$$u = 0 \times \alpha \tag{6}$$

$$\alpha = \text{softmax}\left(W_2 \times e + b_2\right) \tag{7}$$

$$e = \tanh\left(W_1 \times 0 + b_1\right) \tag{8}$$

Where $O = [h_1, h_2, \cdots, h_n]$ represents the hidden layer state matrix of the LSTM. W_1 and W_2 represent the weight matrix of the first and second fully connected layers, respectively. b_1 and b_2 indicate their bias, respectively. First, O is passed through a fully connected neural network, and $\tanh()$ is used as the activation layer to obtain a new hidden layer to express e. The importance of each set of hidden layer state vectors is measured using a second fully connected layer and mapped as a probability distribution using softmax(). The soft attention mechanism is used to sum O by weight α. Finally, the sum of h_n and u is used as the output of the attention mechanism network.

$$y = h_n + u \tag{9}$$

Typically, the attention mechanism network takes u as the output of the network. However, in LSTM, h_n captures most of the information, so h_n is more important than the output at other moments. This method is inspired by the structure of the residual network and chooses y as the output to change the learning objective of the attention mechanism network. Assuming that the true distribution of image information is $H(x)$, when u is used as the output, the goal of the attention mechanism network is to make the output u as close as possible to the true distribution of image information $H(x)$. However, when y is used as the output, the goal of the attention mechanism network is to make the output u fit $H(x) - h_n$ as much as possible, that is, to use the attention mechanism network to collect missing information from n outputs of LSTM to supplement h_n, which ensures the importance of h_n.

The dual metric module proposed in this paper can measure the similarity between node features more robustly. Firstly, the extracted image features are used as node features in the graph neural network, and the similarity between the node features is calculated by the double metric module, and the similarity is used as the edge features of the graph neural network. Then, after h times of updating the node features and edge features of the graph neural network, the prediction probability of each category is output through the loss function.

The dual metric module uses Euclidian distance and cosine function to calculate the similarity between node features [14]. Euclidean distance measures the distance and difference degree by calculating the L2-norm between node features. The smaller the distance, the more similar the node features, and the higher the probability of belonging to the same category. The cosine function calculates the degree of operation, the more similar the distance. The orientation and similarity of nodes in feature space. The greater the cosine similarity, the more similar the node features, and the higher the probability of belonging to the same category. The formula for initializing edge features is as follows:

$$A_{i,j}^0 = f_e\left(\left(v_i^0 - v_j^0\right)^2 + \frac{\left(v_i^0\right)^T v_j^0}{\left\|\left(v_i^0\right)^T\right\| \cdot \left\|v_j^0\right\|}\right) \tag{10}$$

Where v represents the node features extracted by ResNet50. 0 indicates initialization. $v \in R^m$, m indicates the node feature dimension. $A_{i,j}$ represents the edge feature, $f_e : R^{NK} \to R$, and the network structure is shown in Fig. 3, which mainly includes convolution, batch normalization, ReLU activation function and Sigmoid function.

By initializing the node feature and edge feature of the graph neural network, the node feature and edge feature in the graph neural network are updated alternately. The edge feature updating formula is shown in Eq. (11):

$$A_{i,j}^h = f_e\left(\left(\left(v_i - v_j\right)^2 + \frac{v_i^T v_j}{\left\|v_i^T\right\| \cdot \left\|v_j\right\|}\right) \cdot A_{i,j}^{h-1}\right) \tag{11}$$

The characteristics of nodes are updated according to formula (12) to reflect the interaction between nodes and the changes in characteristics.

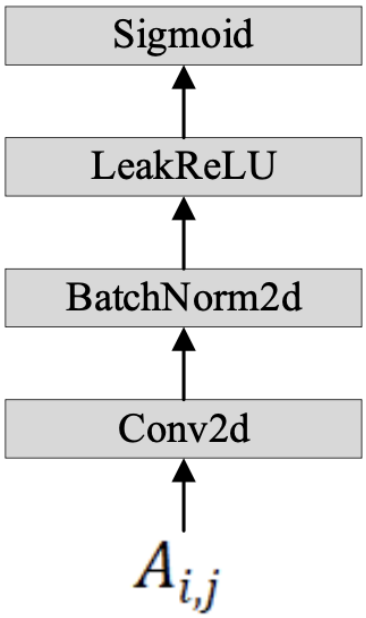

Fig. 3. Edge network structure

$$v_i^h = Gc\left(v_i^h\right) = f_v\left(\sum_{j=1}^{T}\left(A_{i,j}^h \cdot v_j^{h-1}\right), v_j^{h-1}\right) \tag{12}$$

Where $f_v : (R^m, R^m) \to R^m$ contains two blocks of Conv-BN-Relu. $\sum_{j=1}^{T}\left(A_{i,j}^h \cdot v_j^{h-1}\right)$ means that the node feature v_j^{h-1} of the previous layer is multiplied with the adjacency matrix $A_{i,j}^h$, and the result after the multiplication is concatenated with the node feature v_j^{h-1} of the previous layer to prepare for further updating of edge features. Finally, the result of the concatenation is reduced by f dimension to prepare for the updating of node features and edge features of the next layer of graph neural network.

The loss function of the algorithm in this paper is composed of three parts. By adding hyperparameters α and λ, the loss of each part is balanced.

1) The first loss is O_1. Cross entropy loss from real labels to deep classifiers to all shallow classifiers. This value is obtained by calculating the true label value in the dataset with the softmax output of the deep classifier and each shallow classifier, where the loss is:

$$O_1 = S\left(q^0, p\right) + \sum_{i=1}^{3}(1-\alpha)S\left(q^i, p\right) \tag{13}$$

Where S is the cross entropy loss function. q^0 is the output of T = 1 in the deep classifier softmax. q^i is the output of T = 1 in softmax in shallow classifier $i.p$ is the true label value. In this way, knowledge hidden in the data set can be introduced directly from labels to all classifiers.

2) The second loss O_2. The relative entropy loss from the deep classifier to each shallow classifier. The output of softmax of the deep classifier is introduced into softmax of the shallow classifier. The loss is:

$$O_2 = \sum_{i=1}^{3}\alpha K\left(l^i, e\right) \tag{14}$$

Where K is the relative entropy loss and l^i is the output of T = 3 in softmax in shallow classifier $i.e$ is the output of T = 3 in the deep classifier softmax. In this way, the knowledge summarized by the deep classifiers can be transferred to the various shallow classifiers.

3) The third loss is O_3. L2-norm between the output of the hidden layer of the deep classifier and the output of the shallow module of the shallow classifier, where the loss is:

$$O_3 = \sum_{i=1}^{2}\lambda L\left(F_i, f\right) \tag{15}$$

Where L is the L2-norm, F_i is the output of shallow module of shallow classifier i, and f is the output of deep classifier module 4 . In this way, the output of the hidden layer of the deep classifier can be introduced into the shallow module of the 2 shallow classifiers.

4. Experiments and results

4.1. Data sets

This paper conducted experiments on four commonly used image classification datasets, namely, CUB-200-2011 dataset [15], Stanford Dogs dataset [16], Stanford Cars dataset [17] and Mini-Imagenet dataset. The Mini-Imagenet dataset is a subset of the ImageNet dataset [18]. ImageNet is a large image dataset containing millions of images and thousands of categories. Table 1 provides details of these data sets.

4.2. Experiment settings

The experimental environment is Ubuntu 18.04 system, using RTX 3090 TI GPU training. The model training platform uses PyTorch deep learning framework. In the experiment, the size of all images was set to 224×224. At the same time, common data enhancement strategies are used to enhance the data, such as color jitter and horizontal flipping. The model is initialized by loading parameters pre-trained by

ResNet on ImageNet-1K. In this paper, the Adam optimizer is used for parameter optimization, the weight attenuation coefficient is 1×10^5, the initial learning rate is set to 1×10^4, and the learning rate attenuate to 0.1 times of the original after 10,000 iterations. In the test phase,

Table 1. Experimental data set

Dataset	Images	Classes	Train/Val/Test
CUB-200-2011	11800	200	100/50/50
Stanford Dogs	20600	120	70/20/30
Stanford Cars	16200	200	129/17/50
Mini-Imagenet	60000	100	64/16/20

4.3. Experiment settings

The experimental environment is Ubuntu 18.04 system, using RTX 3090 TI GPU training. The model training platform uses PyTorch deep learning framework. In the experiment, the size of all images was set to 224×224. At the same time, common data enhancement strategies are used to enhance the data, such as color jitter and horizontal flipping. The model is initialized by loading parameters pre-trained by ResNet on ImageNet-1K. In this paper, the Adam optimizer is used for parameter optimization, the weight attenuation coefficient is 1×10^5, the initial learning rate is set to 1×10^4, and the learning rate attenuate to 0.1 times of the original after 10,000 iterations. In the test phase, 1000 episodes are selected, and the accuracy of the model evaluation index is 95% confidence.

4.4. Analysis of experimental results

To evaluate the performance of the proposed model, a series of controlled experiments were performed on the CUB-200-2011, Stanford Dogs, and Stanford Cars datasets. Control methods include ICNN [19], HCFNN [20], and FuzzyNet [21].

This paper evaluates the 5-way 1 -shot and 5 -way 5 -shot classification tasks respectively. The convergence curve and loss curve of the model in the CUB-200-2011 data set are shown in Figs. 4 and 5. The accuracy rate increases with the increase of step, and finally becomes stable. At the same time, loss decreases with the increase of step and finally becomes stable.

Table 2 shows the experimental results of this model in CUB-200-2011. As can be seen from the table, in the 5-way 1-shot task, the accuracy of the algorithm proposed in this paper is 11.81%, 6.97% and 5.83% higher than that of ICNN, HCFNN and FuzzyNet respectively. In the 5-way 5-shot task, the accuracy of the proposed algorithm is 2.83%, 1.7% and 0.78% higher than that of ICNN, HCFNN and FuzzyNet, respectively. Experimental results show that the proposed method achieves significant performance improvement on the CUB-200-2011 dataset. By introducing a multi-level attention mechanism, the proposed method can better capture semantic relationships in images and extract deeper features.

Table 2. Accuracy of the CUB-200-2011 data set/%

Method	5-way 1-shot	5-way 5-shot
ICNN	79.28	92.12
HCFNN	84.12	93.25
FuzzyNet	85.26	94.17
Proposed	91.09	94.95

Table 3 shows the experimental results of the proposed model on the Stanford Dogs dataset. As can be seen from the table, in the 5-way 1-shot task, the accuracy of the proposed algorithm is 13.99%, 12.22%, 7.81% higher than that of ICNN, HCFNN and FuzzyNet respectively. In the 5-way 5 -shot task, the accuracy of the proposed algorithm is 11.86%, 10.11% and 7.98% higher than that of ICNN, HCFNN and FuzzyNet, respectively. The proposed algorithm is superior to other methods in both tasks.

Table 3. Accuracy of the Stanford Dogs data set/%

Method	5-way 1-shot	5-way 5-shot
ICNN	71.67	84.28
HCFNN	73.44	86.03
FuzzyNet	77.85	88.16
Proposed	85.66	96.14

Table 4 shows the experimental results of the model in this paper on the Stanford Cars dataset. In the 5-way 1-shot task, the accuracy of the proposed algorithm is 6.15%, 0.94% and 3.35% higher than that of ICNN, HCFNN and FuzzyNet, respectively. It should be noted that if ResNet50 is used as the backbone network, the proposed algorithm outperforms the control method in both tasks. This shows that the proposed method has achieved some performance improvement on the data set. Although the accuracy of the proposed algorithm is slightly lower than that of the control method in the 5-way 5-shot task, the proposed method is still effective for image classification tasks.

Table 4. Accuracy of the Stanford Cars data set/%

Method	5-way 1-shot	5-way 5-shot
ICNN	85.38	92.61
HCFNN	90.59	93.47
FuzzyNet	88.18	95.83
Proposed	91.53	94.62

Table 5 shows the experimental results of the model

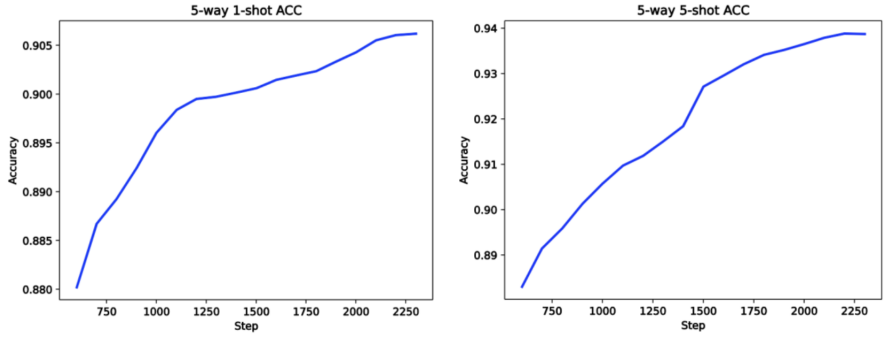

Fig. 4. Model converegence on the CUB-200-2011dataset

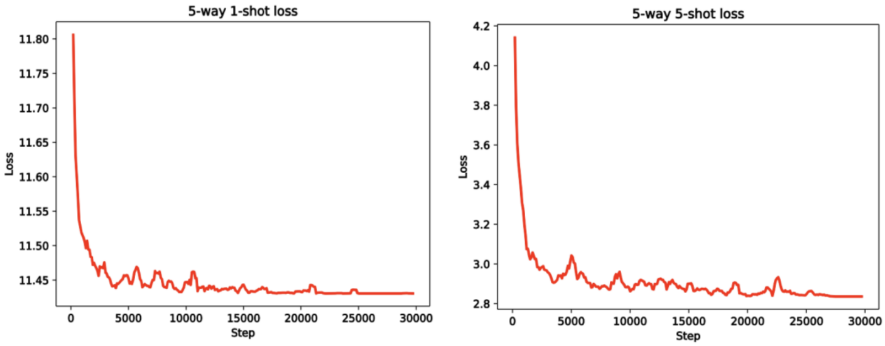

Fig. 5. Model loss on the CUB-200-2011 dataset

in this paper on the Mini-Imagenet dataset. In the 5-way 1-shot task, the accuracy of the algorithm in this paper is significantly improved compared with that of ICNN, HCFNN and FuzzyNet. In the 5-way 5-shot task, the accuracy of the proposed algorithm in this paper is 1.79% higher than that of HCFNN and 0.02% lower than that of FuzzyNet. This shows that the proposed method achieves some performance improvement on the Mini-Imagenet data set.

Table 5. Accuracy of the Mini-Imagenet data set/%

Method	5-way 1-shot	5-way 5-shot
ICNN	65.71	82.68
HCFNN	66.37	83.24
FuzzyNet	68.09	85.03
Proposed	72.47	85.01

4.5. Ablation experiment

In this paper, three sets of ablation experiments were performed on the CUB-200-2011 dataset to verify the validity of the proposed model. The first set of experiments, using the ResNet50 network feature extraction module way; The second set of experiments used ResNet50 network and L2 norm measurement. The third set of experiments used the feature extraction module and attention mechanism approach of the ResNet50 network. The experimental results are shown in Table 6.

The experimental results show that in 5-way 1-shot tasks, the classification accuracy of the proposed strategy is 25.91% higher than that of the feature extraction network module of ResNet50 alone. In the 5-way 5 -shot task, classification accuracy improved by 3.76%. The feature extraction module of the network in this paper can better capture the semantic relations in images and extract deeper features.

5. Conclusions

In this paper, ResNet50 is combined with attention mechanism and applied to image classification task. By taking advantage of ResNet50's powerful feature representation capabilities, the model is able to capture more global and richer image information. In order to solve the accuracy problem which may be caused by using a single similarity measure for edge features in neural networks, this paper introduces an edge-double measure module. The module calculates the similarity between node features by calculating edge features, so as to improve the accuracy of

Table 6. Ablation results/%

Numb.	ResNet50	L2 norm	attention	5-way 1-shot	5-way 5-shot
1	√			75.61	91.37
2	√	√		88.66	93.49
3	√		√	91.52	95.13

similarity measurement. The model was tested on three datasets: CUB-200-2011, Stanford Cars and Stanford Dogs. The experimental results show that the proposed model is superior to other methods and improves the accuracy of image classification.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. U20A20197, 61973063, Liaoning Key Research and Development Project 2020JH2/10100040, Natural Science Foundation of Liaoning Province 2021-KF-12-01 and the Foundation of National Key Laboratory OEIP-O-202005.

References

[1] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle, (2022) *"Unlocking high-accuracy differentially private image classification through scale, 2022"* **arXiv preprint arXiv:2204.13650**: DOI: 10.48550/arXiv.2204.13650.

[2] X. Meng, X. Wang, S. Yin, and H. Li, (2023) *"Few-shot image classification algorithm based on attention mechanism and weight fusion"* **Journal of Engineering and Applied Science *70***(1): 1–14. DOI: 10.1186/s44147-023-00186-9.

[3] L. Zhan, W. Li, and W. Min, (2023) *"FA-ResNet: Feature affine residual network for large-scale point cloud segmentation"* **International Journal of Applied Earth Observation and Geoinformation *118***: 103259. DOI: 10.1016/j.jag.2023.103259.

[4] M. Ji, G. Peng, S. Li, F. Cheng, Z. Chen, Z. Li, and H. Du, (2022) *"A neural network compression method based on knowledge-distillation and parameter quantization for the bearing fault diagnosis"* **Applied Soft Computing *127***: 109331. DOI: 10.1016/j.asoc.2022.109331.

[5] S. Lin, R. Ji, C. Chen, D. Tao, and J. Luo, (2018) *"Holistic cnn compression via low-rank decomposition with knowledge transfer"* **IEEE transactions on pattern analysis and machine intelligence *41***(12): 2889–2905. DOI: 10.1109/TPAMI.2018.2873305.

[6] X.-L. Zhang, B.-C. Du, Z.-C. Luo, and K. Ma, (2022) *"Lightweight and efficient asymmetric network design for real-time semantic segmentation"* **Applied Intelligence *52***(1): 564–579. DOI: 10.1007/s10489-021-02437-9.

[7] L. Wang and K.-J. Yoon, (2021) *"Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks"* **IEEE transactions on pattern analysis and machine intelligence *44***(6): 3048–3068. DOI: 10.1109/TPAMI.2021.3055564.

[8] T.-B. Xu and C.-L. Liu, (2020) *"Deep neural network self-distillation exploiting data representation invariance"* **IEEE Transactions on Neural Networks and Learning Systems *33***(1): 257–269. DOI: 10.1109/TNNLS.2020.3027634.

[9] Y. Cui, Y. An, W. Sun, H. Hu, and X. Song, (2020) *"Lightweight attention module for deep learning on classification and segmentation of 3-D point clouds"* **IEEE Transactions on Instrumentation and Measurement *70***: 1–12. DOI: 10.1109/TIM.2020.3013081.

[10] L. Teng, Y. Qiao, M. Shafiq, G. Srivastava, A. R. Javed, T. R. Gadekallu, and S. Yin, (2023) *"FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction"* **IEEE Transactions on Network and Service Management *20***(2): 1529–1542. DOI: 10.1109/TNSM.2023.3273991.

[11] A. Jisi, S. Yin, et al., (2021) *"A new feature fusion network for student behavior recognition in education"* **Journal of Applied Science and Engineering *24***(2): 133–140. DOI: 10.6180/jase.202104_24(2).0002.

[12] C. Zhang, L. Wang, S. Cheng, and Y. Li, (2022) *"SwinSUNet: Pure transformer network for remote sensing image change detection"* **IEEE Transactions on Geoscience and Remote Sensing *60***: 1–13. DOI: 10.1109/TGRS.2022.3160007.

[13] J. Zhu, Y. Tan, R. Lin, J. Miao, X. Fan, Y. Zhu, P. Liang, J. Gong, and H. He, (2022) *"Efficient self-attention mechanism and structural distilling model for Alzheimer's disease diagnosis"* **Computers in Biology and Medicine *147***: 105737. DOI: 10.1016/j.compbiomed.2022.105737.

[14] J. Qu, Y. Xu, W. Dong, Y. Li, and Q. Du, (2021) *"Dual-branch difference amplification graph convolutional network for hyperspectral image change detection"* **IEEE Transactions on Geoscience and Remote Sensing *60***: 1–12. DOI: 10.1109/TGRS.2021.3135567.

[15] H. Li, X. Zhang, Q. Tian, and H. Xiong. "Attribute mix: Semantic data augmentation for fine grained recognition". In: *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE. 2020, 243–246. DOI: 10.1109/VCIP49819.2020.9301763.

[16] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. "Novel dataset for fine-grained image categorization: Stanford dogs". In: *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*. **2**. 1. Citeseer. 2011.

[17] T. Kramberger and B. Potočnik, (2020) *"LSUN-Stanford car dataset: enhancing large-scale car image datasets using deep learning for usage in GAN training"* **Applied Sciences *10***(14): 4913. DOI: 10.3390/app10144913.

[18] B. Oreshkin, P. Rodríguez López, and A. Lacoste, (2018) *"Tadam: Task dependent adaptive metric for improved few-shot learning"* **Advances in neural information processing systems *31***:

[19] W. Zhou, H. Wang, and Z. Wan, (2022) *"Ore image classification based on improved CNN"* **Computers and Electrical Engineering *99***: 107819. DOI: 10.1016/j.compeleceng.2022.107819.

[20] X. Ning, W. Tian, Z. Yu, W. Li, X. Bai, and Y. Wang, (2022) *"HCFNN: high-order coverage function neural network for image classification"* **Pattern Recognition *131***: 108873. DOI: 10.1016/j.patcog.2022.108873.

[21] V. Narayan, P. K. Mall, S. Awasthi, S. Srivastava, and A. Gupta. "FuzzyNet: Medical Image Classification based on GLCM Texture Feature". In: *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*. IEEE. 2023, 769–773. DOI: 10.1109/AISC56616.2023.10085348.