Multi-Branch Network for Few-shot Learning

Kai Ren*, Zijie Guo*, Zhimin Zhang*, Rui Zhu†, Xiaoxu Li*
* Lanzhou University of Technology, Lanzhou, China
E-mail: lixiaoxu@lut.edu.cn
† Faculty of Actuarial Science and Insurance, Bayes Business School, City, University of London, UK
E-mail: rui.zhu@city.ac.uk

***Abstract*—Few-shot learning aims provide precise predictions for unseen data through learning from only one or few labelled samples of each class. However, it often suffers from the overfitting problem because of insufficient training data. In this paper, we propose a novel metric-based few-shot learning method, multi-branch network (MBN), with a new data augmentation module to improve the generalization ability of the model. Specifically, we generate different types of noise contaminated data through multiple branches in the network to simulate the real-world scenarios when noisy images are obtained. Following this novel data augmentation module, the feature embedding and similarities between the support and query samples are learned simultaneously through the embedding and metric modules, respectively. Moreover, to consider more details in the feature maps, we propose to utilize the average-pooling layer in the metric module rather than the commonly adopted max-pooling layer. The network is trained from end to end by the Kullback-Leibler (KL) divergence, to minimize the difference between the distributions of the ground truths and predictions. Extensive experiments on Standford-Dogs, Standford-Cars, CUB-200-2011 and mini-ImageNet in the 1-shot and 5-shot tasks demonstrate the superior classification performance of MBN.**

I. Introduction

Although vast amount of advanced machine learning algorithms have been developed, machines are still not as accurate as humans [1], [2], [3]. The well-performed algorithms usually have to be trained by a large amount of high-quality data, while human beings can learn complex tasks quickly through very few examples and even noisy ones. Therefore, few-shot learning [1], which aims to achieve precise recognition of unseen data through only one or few labelled samples of each class, has received wide attention recently.

Metric-based methods are one attractive category in few-shot learning, which aims to classify a query sample based on its similarities or dissimilarities to the support samples. In early studies, pre-defined metrics are adopted, such as Euclidean distance [4], cosine similarity [5] and L1 distance [6]. Recently, more and more literature proposes to learn data-adaptive metrics; one famous example is the relation network (RelationNet) [7] which involves the relation module to learn the relationship between the query and support samples. Rather than measuring the similarities or dissimilarities between the query sample and all support samples, prototype-based algorithms are proposed to construct class prototypes from the support samples and classify the query sample by its similarities or dissimilarities to the class prototypes. For example, prototype network (PrototypeNet) [4] calculates the class prototype as the sample mean of the support set of that class. Moreover, feature embeddings and metrics can be learned simultaneously through the network. Position-aware relation network [8] aims to properly learn the similarities between semantically related-objects of two images by utilizing a deformable feature extractor and a dual correlation attention mechanism. To further consider the discriminative information between classes, Li et al. [9] propose to learn task-relavant features via the category traversal module.

Nowadays, metrics are learned through more advanced networks. Nguyen et al. propose [10] to use the Euclidean distance and square root of the norm distance to constrain the modular length of the feature to the modular length of the prototype. Li et al. [11] propose an asymmetric distribution metric network, which includes two types of metrics to measure different characteristics of the images. In addition, they also propose the task-aware contrast metric strategy that acts as a plugin to enhance the measure function.

However, few-shot learning algorithms often suffer from the overfitting problem because of the small amount of training data available. Data augmentation is an efficient approach to solve this problem. For example, new training data can be generated via geometric transformation [12], [13] and color transformation [14] of the original data. However, existing data augmentation strategies for few-shot learning do not consider the real-world scenario when noises present. In this paper, we propose a novel method to generate new training data via two parallel branches in the network to incorporate Gaussian and salt-and-pepper noises and we name this method multi-branch network (MBN). We expect that MBN can alleviate the overfitting problem and is more reliable when classifying contaminated data.

We illustrate the effectiveness of this new data augmentation strategy through redesigning the architecture of RelationNet that can learn the feature embedding and metric simultaneously. Besides the new data augmentation strategy, we also propose to replace the max-pooling layer in the relation module of RelationNet by the average-pooling layer, because max-pooling only retains the texture features while average-pooling considers all information and does not ignore potentially important information from feature maps.

To sum up, the contributions of this paper are as follows:

1) We propose a new data augmentation strategy for few-

shot learning, by generating noise-contaminated data through parallel branches in the network, to enhance the generalization ability of the model.

2) We design a new metric module with the aid of the average-pooling layer to avoid loss of important information for classification.

3) Experiments on four benchmark datasets for few-shot learning demonstrate the superior classification performance of MBN.

II. Related Work

Our work draws inspiration from rich literature on data augmentation, metric-based few-shot learning and the loss functions to train neural networks.

A. Data Augmentation

The commonly used methods in data augmentation are deformation [12], [13], such as clipping, filling and horizontal flipping, generating more training samples [15] and pseudo labels [16]. Zhang et al. [17] expand the training dataset by splicing the foreground and background of different images to generate more composite images. Gidaris et al. [18] rotate the original image at different angles and calculated the rotation angle and classification task, respectively, through the feature extraction network. Das et al. [19] propose a new fine-tuning method to improve the generalization ability based on contrast learning, which reuses the unlabelled instances in the basic domain as distractors. Hybrid feature space learning method [20] utilizes hybrid models to model base classes by training feature extractors online and learning parameters of hybrid models simultaneously.

Different from these methods, the data augmentation module in MBN can simulate the real-world scenarios when different types of noises present. This makes our algorithm more reliable when classifying noisy images.

B. Metric-Based Few-Shot Learning

Besides the widely adopted metrics, such as Euclidean distance and cosine similarity, more advanced metrics are utilized for better classification performance. Zhang et al. [21] propose a variational Bayesian framework and used KL divergence to measure the distance between samples. Their framework calculates the predicted probability of a query sample by estimating the sample distribution of each class. DeepEMD [22] utilizes earth mover's distance to measure the structural similarity between two images that are represented by their building blocks. Rather than pre-defined metrics, some studies propose to learn the metrics through training data automatically. For example, RelationNet learns the metric and feature embedding via a convolutional neural network module. The metric-based meta-learning method COMET [23] learns concept-specific metrics and aggregates decisions from different concept learners for final decision. Li et al. [24] propose the bi-similarity metric network (BSNet) by using two different similarity measures in the metric module to learn distinct characteristics.

To the best of our knowledge, most of the metric modules in literature is constructed by the max-pooling layer which only extracts the extreme values from the feature maps and ignores all other information that can be valuable for classification. Thus we propose to utilize average-pooling in the metric module of MBN to obtain good overall representations of the feature maps without the loss of potentially vital information.

C. Loss Function

Neural networks are usually trained by minimizing a loss function that measures the differences between predictions and ground truths. Li et al. [25] propose the dual cross-entropy loss function by adding a penalty term that can limit the probability of being assigned to the wrong classes and alleviate the vanishing gradient problem. Wu et al. [26] propose a new triplet loss function that can pull similar samples together while push dissimilar samples apart. MeTAL, a meta-learning framework with task adaptive loss function [27], learns a loss function that can be modified in the inner-loop according to the current task requirements, so that the loss function can fit to each task and improve the generalization ability of the model.

Different from the above methods, we propose to minimize the Kullback–Leibler (KL) divergence [21] in MBN, to make the distributions of the predictions closely match that of the ground truths.

III. Methodology

In this section, we first define the problem of few-shot learning in section III-A. We then introduce the process of RelationNet in section III-B. We finally describe the details of the proposed MBN in section III-C.

A. Problem Definition

In few-shot learning, the dataset is usually divided to a training set T_{train}, a test set T_{test} and a validation set T_{val}, which are mutually exclusive to each other. Their corresponding label sets are denoted as $\mathcal{T}_{\text{train}}$, $\mathcal{T}_{\text{test}}$ and $\mathcal{T}_{\text{val}}$, respectively.

During each training iteration, C classes are randomly selected from the training set and K images from each class are randomly selected to form the support set $S = \{(\mathbf{x}_e, y_e), y_e \in \mathcal{T}_{\text{train}}\}_{e=1}^{m}$, where $\mathbf{x}_e$ is the e-th image in the support set and y_e is its label. Then, a fraction of the rest training samples in each class are randomly selected as the query set $Q = \{(\mathbf{x}_p, y_p), y_p \in \mathcal{T}_{\text{train}}\}_{p=1}^{n}$, where $\mathbf{x}_p$ is the p-th image in the query set and y_p is its label. This sampling method is often called "C-way K-shot" in few-shot learning.

B. Relation Network

Relation network (RelationNet) [7] is a simple yet effective metric-based method for few-shot learning and zero-shot learning. It consists of two modules, the embedding module to extract features from the support and query samples and the relation module to evaluate the relation scores between them. The details of RelatioNet are described below.

1) *The embedding module f_θ*:

$$\mathbf{t}_e = f_\theta\left(\mathbf{x}_e\right) \in \mathbb{R}^{c\times h\times w},\ e \in \{1,\ldots,\mathrm{m}\}, \tag{1}$$

$$\mathbf{h}_p = f_\theta\left(\mathbf{x}_p\right) \in \mathbb{R}^{c\times h\times w},\ p \in \{1,\ldots,n\}, \tag{2}$$

where c, h and w indicate the number of channels and the height and width of feature maps, respectively. The embedding module consists of four convolutional blocks, each of which is composed of 64 filters, and each filter is composed of a 3×3 convolution, a batch normalization and ReLU nonlinear layer. Each of the first two convolutional blocks of the four convolutional blocks are followed by a 2×2 max-pooling layer, while the last two convolutional blocks are not followed by any pooling layers.

2) *Feature maps concatenation*: The features of the support and query samples extracted by the embedding module are concatenated along the direction of the channels to obtain the support-query feature pairs:

$$\mathbf{J}_{(e,p)} = F(f_\theta(\mathbf{x}_e), f_\theta(\mathbf{x}_p)) \in \mathbb{R}^{2c\times h\times w}, \tag{3}$$

where F denotes the concatenation operation.

3) *The relation module g_ψ*:

$$R_{(e,p)} = g_\psi(F(f_\theta(\mathbf{x}_e), f_\theta(\mathbf{x}_p))), \tag{4}$$

where $R_{(e,p)}$ is the relation score between the support sample $\mathbf{x}_e$ and the query sample $\mathbf{x}_p$. The relation module consists of two convolutional blocks and two fully connected layers. The compositions of the convolutional blocks here are the same as those of the embedding module.

4) *The loss function L_{MSE}*: The following mean-squared error(MSE) loss function is used to train the network:

$$\mathrm{L_{MSE}} = \sum_{e=1}^{m}\sum_{p=1}^{n}\left(\mathrm{R}_{(e,p)} - \mathbf{1}\left(y_e == y_p\right)\right)^2, \tag{5}$$

where $\mathbf{1}$ is the identity function.

C. Multi-Branch Network

In order to reduce the risk of overfitting and improve the generalization ability of the model, we propose to enlarge the training set by involving an additional data augmentation module. Different from the existing works on data augmentation, we focus on simulating scenarios when the images are contaminated by different types of noises, which is commonly seen in real-world data. In this paper, we generate support samples with two types of noises, Gaussian noise and salt-and-pepper noise, to form two branches for the data augmentation module, with the third branch containing the original support samples. Because of this multi-branch data augmentation strategy, we name our method multi-branch network (MBN). Following the data augmentation module, we incorporate an embedding module that is the same as RelationNet, and a metric module with average-pooling layers to consider more details of the feature maps. The architecture of MBN is illustrated in Fig. 1 via an example of the 3-way 1-shot learning process. The details of each module of MBN are described below.

1) *The data augmentation module, $d_{\varphi 1}, d_{\varphi 2}$ and $d_{\varphi 3}$*: Contaminated support samples are generated by adding Gaussian noises and salt-and-pepper noises to the original support samples:

$$d_{\varphi 1}(\mathbf{x}_e) = \mathbf{x}_e + \boldsymbol{\delta}_e^G, \tag{6}$$

$$d_{\varphi 2}(\mathbf{x}_e) = \mathbf{x}_e + \boldsymbol{\delta}_e^{SP}, \tag{7}$$

where $\boldsymbol{\delta}_e^G$ and $\boldsymbol{\delta}_e^{SP}$ are the Gaussian noises and salt-and-pepper noises added to $\mathbf{x}_e$, respectively. The third branch of the data augmentation module is the original support samples:

$$d_{\varphi 3}(\mathbf{x}_e) = \mathbf{x}_e. \tag{8}$$

Note that the data augmentation module only process the support samples and the query samples are not involved in this stage.

The dimensions of support samples are $[3, 84, 84]$, where 3 is the number of channels in the image, and the two 84's are the height and width of the image, respectively. The dimensions are kept the same after the data augmentation process.

2) *The embedding module f_θ and feature maps concatenation*: these two steps are the same as those in RelationNet. Here we process the three branches separately; that is, for each branch of support samples, we extract its features and concatenate them with the extracted features of the query sample.

3) *The metric module h_ϕ*: The compositions of the metric module is the same as the relation module in RelationNet, except that we replace the 2×2 max-pooling layer by a 2×2 average-pooling layer. This is because the max-pooling layer only retains texture features and compresses most of the information in feature maps, and a lot of potentially valuable information for classification may be ignored. On the contrary, the average-pooling layer considers all values and can include more details from the feature maps. Fig. 2 shows the structure of the metric module of MBN. The similarities between the support sample and query sample for each branch are calculated as follows:

$$R_{(e,p)-1} = h_\phi\left(\mathrm{F}\left(f_\theta(d_{\varphi_1}(\mathbf{x}_e)), f_\theta\left(\mathbf{x}_p\right)\right)\right), \tag{9}$$

$$R_{(e,p)-2} = h_\phi\left(\mathrm{F}\left(f_\theta(d_{\varphi_2}(\mathbf{x}_e)), f_\theta\left(\mathbf{x}_p\right)\right)\right), \tag{10}$$

$$R_{(e,p)-3} = h_\phi\left(\mathrm{F}\left(f_\theta(d_{\varphi_3}(\mathbf{x}_e)), f_\theta\left(\mathbf{x}_p\right)\right)\right). \tag{11}$$

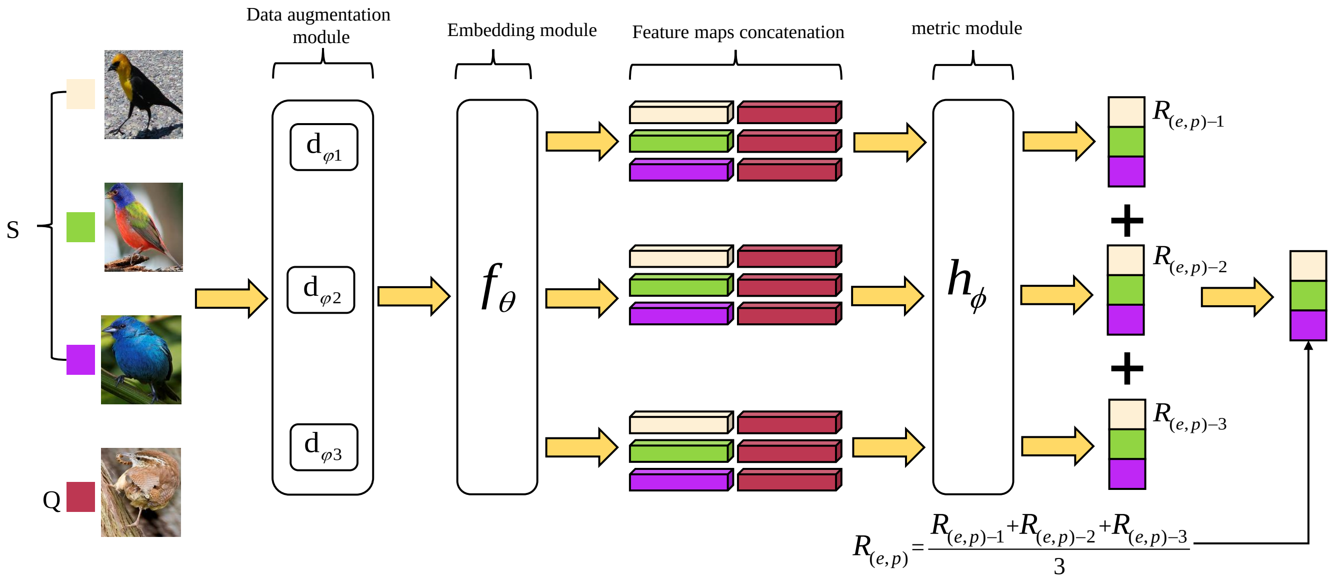

Fig. 1. An illustration of the architecture of MBN. The proposed network consists of the data augmentation module, embedding module and metric module. This example illustrates the 3-way 1-shot learning process.

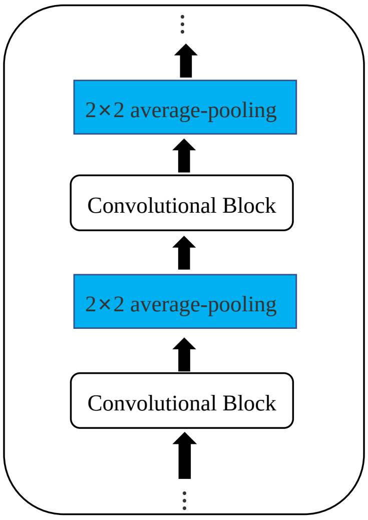

Fig. 2. The structure of the metric module of MBN.

These three similarities are aggregated by their average as the final similarity between the support sample $\mathbf{x}_e$ and query sample $\mathbf{x}_p$:

$$R_{(e,p)} = \frac{R_{(e,p)-1} + R_{(e,p)-2} + R_{(e,p)-3}}{3}. \tag{12}$$

4) *The loss function* L_{KL}: We propose to use the KL divergence [21] to train our model, which measures how the distributions of the true labels are different from the similarities:

$$L_{KL} = \sum_{p=1}^{n}\sum_{e=1}^{m}\left(y_p \log \frac{y_p}{R_{(e,p)}}\right). \tag{13}$$

IV. Experiments

In this section, we provide details of the datasets to evaluate MBN and the experiment settings in sections IV-A and IV-B, respectively. In section IV-C, we compare the classification performances of MBN with nine state-of-the-art few-shot learning methods. Finally, we test the efficiency of each module in MBN in section IV-D.

A. Dataset

We choose four commonly used publicly available datasets in few-shot learning in our experiments: mini-ImageNet [5], Stanford-Dogs (Dogs) [28], Stanford-Cars (Cars) [29] and CUB-200-2011 (CUB) [30]. In the rest of the paper, we use the short notations in the brackets to denote the first three datasets.

The Dogs data contain 120 dog categories and the total number of images is 20,580. We randomly divide them to a training set with 10,337 images of 60 categories, a validation set with 5,128 images of 30 categories and a test set with 5,115 images of 30 categories.

The Cars data consist of 192 classes of cars with a total of 16,185 images. We randomly divide them to a training set with 8,023 images of 98 classes, a validation set with 4,059 images of 49 classes and a test set with 4,103 images of 49 classes.

The CUB data have 200 species of birds with a total of 11,788 images. We randomly divide them to a training set with 8,023 images of 98 classes, a validation set with 4,059 images of 49 classes and a test set with 4,103 images of 49 classes.

The mini-ImageNet data are a subset of ImageNet, which contain 60,000 color images in 100 classes. We randomly

TABLE I
THE 1-SHOT AND 5-SHOT CLASSIFICATION PERFORMANCES ON THE DOGS, CARS AND CUB DATA. THE HIGHEST ACCURACY IN EACH COLUMN IS DENOTED BY BOLD FONT. WE OBTAIN THE RESULTS OF THE METHODS MARKED WITH "# " BY OUR OWN REPLICATIONS BASED ON THE CODES IN LITERATURE. THE "-" ENTRY MEANS THAT NO EXPERIMENT HAS BEEN CONDUCTED IN LITERATURE.

Method	5-Way Accuracy(%)					
	Dogs		Cars		CUB	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet(2016)	46.20± 0.88	62.50± 0.73	44.63± 0.77	64.70± 0.72	60.17± 0.86	74.55± 0.72
PrototypeNet(2017)	45.25± 0.80	61.50± 0.75	48.45± 0.23	71.40± 0.16	62.89± 0.26	70.60± 0.17
MAML(2017)	47.10± 0.80	62.45± 0.81	48.27± 0.80	65.35± 0.73	55.87± 0.97	72.39± 0.75
RelationNet#(2018)	47.35± 0.92	65.65± 0.64	46.24± 0.89	68.42± 0.78	61.89± 0.95	78.16± 0.66
DN4#(2019)	45.11± 0.70	63.56± 0.65	59.83± 0.87	**88.60± 0.48**	-	-
DeepEMD#(2020)	46.76± 0.47	65.54± 0.65	**61.53± 0.23**	72.90± 0.38	64.18± 0.50	**80.15± 0.70**
LRPABN#(2020)	45.70± 0.77	60.90± 0.60	60.38± 0.76	73.49± 0.58	63.60± 0.75	76.24± 0.66
BSNet#(2021)	43.90± 0.80	62.65± 0.65	44.36± 0.83	63.52± 0.78	55.71± 0.95	76.24± 0.63
MixtFSL#(2021)	43.90± 0.70	64.33± 0.63	44.36± 0.82	59.53± 0.80	53.51± 0.81	73.20± 0.75
MBN(Ours)	**50.15± 0.92**	**67.13± 0.73**	60.55± 0.97	76.35± 0.74	**64.85± 0.93**	79.20± 0.64

divide them to a training set with 38,400 images of 64 classes, a validation set with 9,600 images of 16 classes and a test set with 12,000 images of 20 classes.

B. Experiment settings

We carry out two few-shot learning settings in the experiments: 5-way 1-shot and 5-way 5-shot learning. The number of query samples for each image in the experiment is set to 16. We adopt Conv4 and set the number of channels to 64; thus, the extracted feature is of size $64\times19\times19$. The initial learning rate was set to 10^{-3} during the training process which starts from scratch with the Adam optimizer.

To demonstrate the validity of the proposed method, nine state-of-the-art methods are selected as baselines for comparison: PrototypeNet [4], MatchingNet [5], BSNet [24], RelationNet [7], MixtFSL [20], DeepEMD [22], deep nearest neighbor neural network (DN4) [31], LRPABN [32] and MAML [33].

C. Experiment results

We report the classification results on the Dogs, Cars and CUB data in Table I. It is obvious that MBN outperforms all baselines on the Dogs data for both 1-shot and 5-shot tasks. MBN also beats the state-of-the-art methods for 1-shot on the CUB data. For other tasks and datasets, MBN can still provide competitive classification performances compared with baselines.

The experiments in Table I are conducted on fine-grained data. In order to prove that our method is also effective on coarse-grained data, we evaluate the classification performance of MBN on mini-ImageNet. The classification accuracies in Table II clearly demonstrate the effectiveness of MBN on coarse-grained data.

D. Ablation Studies

Here we conduct ablation studies to test the effectiveness of each module of MBN on the CUB data and the results are reported in Table IV. The compositions of the methods in this section are summarised in Table III. We can observe from the results in Table IV that MBN performs the best on the CUB data for both 1-shot and 5-shot tasks, which shows the effectiveness of our proposed novel modules.

TABLE II
THE 1-SHOT AND 5-SHOT CLASSIFICATION PERFORMANCES ON THE MINI-IMAGENET DATASET. THE HIGHEST ACCURACY IN EACH COLUMN IS DENOTED BY BOLD FONT.

Method	mini-ImageNet	
	1-shot	5-shot
MatchingNet(2016)	48.23± 0.78	63.27± 0.68
PrototypeNet (2017)	44.13± 0.85	65.95± 0.72
MAML(2017)	46.77± 0.80	62.75± 0.78
RelationNet(2018)	49.25± 0.89	64.84± 0.70
MBN(Ours)	**50.15± 0.83**	**66.06± 0.67**

TABLE III
THE COMPOSITIONS OF THE METHODS FOR ABLATION STUDIES.

	GRN	SRN	HRN	HKLRN	GHKL
Gaussian noise	✓				✓
Salt & pepper noise		✓			
Relation module	✓	✓			
Metric module			✓	✓	✓
MSE	✓	✓	✓		
KL				✓	✓

TABLE IV
THE CLASSIFICATION PERFORMANCES OF THE METHODS IN ABLATION STUDY ON THE CUB DATASET. NOTE: BOLD FONT IS THE OPTIMAL VALUE FOR EACH COLUMN. THE HIGHEST ACCURACY IN EACH COLUMN IS DENOTED BY BOLD FONT.

Method	CUB	
	1-shot	5-shot
GRN	63.14± 0.90	78.70± 0.57
SRN	63.07± 0.89	78.82± 0.60
HRN	63.80± 0.91	78.94± 0.55
HKLRN	64.31± 0.87	79.06± 0.63
GHKL	64.15± 0.93	79.12± 0.64
MBN(Ours)	**64.85± 0.93**	**79.20± 0.64**

V. CONCLUSION

We propose a novel metric-based few-shot learning method, MBN, which consists of three parts: the data augmentation module, embedding module and metric module. By introducing the data augmentation module, the problem of over-fitting is alleviated and the generalization ability of the model is improved. By using the average-pooling layer in the metric module, the overall characteristics of the images are considered to avoid loss of important information for classification. Experimental results demonstrate the superior classification performance of MBN on the Dogs, Cars, CUB and mini-ImageNet datasets, compared with the state-of-the-art methods.

References

[1] Xiaoxu Li, Zhuo Sun, Jing-Hao Xue, and Zhanyu Ma. A concise review of recent few-shot meta-learning methods. *Neurocomputing*, 456:463–468, 2021.

[2] Jun Shu, Zongben Xu, and Deyu Meng. Small sample learning in big data era. *arXiv preprint arXiv:1808.04572*, 2018.

[3] Jiang Lu, Pinghua Gong, Jieping Ye, and Changshui Zhang. Learning from very few samples: A survey. *arXiv preprint arXiv:2009.02653*, 2020.

[4] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[5] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[6] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[7] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[8] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. PARN: Position-aware relation networks for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6659–6667, 2019.

[9] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2019.

[10] Van Nhan Nguyen, Sigurd Løkse, Kristoffer Wickstrøm, Michael Kampffmeyer, Davide Roverso, and Robert Jenssen. SEN: A novel feature normalization dissimilarity measure for prototypical few-shot learning networks. In *European Conference on Computer Vision*, pages 118–134. Springer, 2020.

[11] Wenbin Li, Lei Wang, Jing Huo, Yinghuan Shi, Yang Gao, and Jiebo Luo. Asymmetric distribution measure for few-shot learning. *arXiv preprint arXiv:2002.00153*, 2020.

[12] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. *Advances in Neural Information Processing Systems*, 28, 2015.

[13] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *Advances in Neural Information Processing Systems*, 30, 2017.

[14] Ze Lu, Xudong Jiang, and Alex Kot. Enhance deep learning performance in face recognition. In *2017 2nd International conference on image, vision and computing (ICIVC)*, pages 244–248. IEEE, 2017.

[15] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Augmenting image classifiers using data augmentation generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 594–603. Springer, 2018.

[16] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in Neural Information Processing Systems*, 29, 2016.

[17] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2770–2779, 2019.

[18] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8059–8068, 2019.

[19] Rajshekhar Das, Yu-Xiong Wang, and José MF Moura. On the importance of distractors for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9030–9040, 2021.

[20] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Mixture-based feature space learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9041–9051, 2021.

[21] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1685–1694, 2019.

[22] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020.

[23] Kaidi Cao, Maria Brbic, and Jure Leskovec. Concept learners for few-shot learning. *arXiv preprint arXiv:2007.07375*, 2020.

[24] Xiaoxu Li, Jijie Wu, Zhuo Sun, Zhanyu Ma, Jie Cao, and Jing-Hao Xue. BSNet: Bi-similarity network for few-shot fine-grained image classification. *IEEE Transactions on Image Processing*, 30:1318–1331, 2020.

[25] Xiaoxu Li, Liyun Yu, Dongliang Chang, Zhanyu Ma, and Jie Cao. Dual cross-entropy loss for small-sample fine-grained vehicle classification. *IEEE Transactions on Vehicular Technology*, 68(5):4204–4212, 2019.

[26] Fangyu Wu, Jeremy S Smith, Wenjin Lu, Chaoyi Pang, and Bailing Zhang. Attentive prototype few-shot learning with capsule network-based embedding. In *European Conference on Computer Vision*, pages 237–253. Springer, 2020.

[27] Sungyong Baik, Janghoon Choi, Heewon Kim, Dohee Cho, Jaesik Min, and Kyoung Lee. Meta-learning with task-adaptive loss function for few-shot learning. pages 9445–9454, 10 2021.

[28] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.

[29] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

[30] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-UCSD birds-200-2011 dataset. 2011.

[31] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[32] Huaxi Huang, Junjie Zhang, Jian Zhang, Jingsong Xu, and Qiang Wu. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. *IEEE Transactions on Multimedia*, 23:1666–1680, 2020.

[33] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.