Few-Shot Fine-Grained Image Classification via Multi-Frequency Neighborhood and Double-Cross Modulation

Hegui Zhu, Zhan Gao, Jiayi Wang, and Chengqing Li

Abstract—Traditional fine-grained image classification typically relies on large-scale training samples with annotated ground truth. However, some fine-grained categories in the real world have few available images, and the existing few-shot models have difficulty in distinguishing the subtle differences among them. Moreover, the intra-class distances between some fine-grained categories may be very large, but the inter-class ones are small, which makes the distinguishing features of each category are different for distinct tasks. To solve the challenges, we propose a novel network (FicNet) using multi-frequency neighborhood (MFN) and double-cross modulation (DCM). MFN captures the multi-frequency structure representation that is irrelevant to the background by integrating the spatial and frequency domain information, and then reduces the intra-class distance. DCM modulates the representation by global context and inter-class relationships, which enables both support and query features to have complete targets and respond to the same parts, and then accurately identify subtle inter-class differences. Comprehensive experiments on three fine-grained benchmark datasets for two few-shot tasks verify that FicNet has excellent performance compared to the state-of-the-art methods. Notably, it can obtain classification accuracy 93.17% and 95.36% on datasets “Caltech-UCSD Birds” and “Stanford Cars”, respectively, surpassing the benchmarks set by general fine-grained image classification methods.

Index Terms—Few-shot learning, feature modulation, fine-grained image classification, multi-frequency component, weighted neighborhood.

I. INTRODUCTION

FINE-GRAINED image classification aims to accurately distinguish different sub-categories belonging to the same entry-level category, such as different kinds of birds, flowers, and cars. Particularly, with the development of deep learning [1], [2], some fine-grained classification methods based on the neural network attracted much attention in various application scenarios, such as automatic biodiversity monitoring [3] and intelligent detection [4]. Traditionally, fine-grained classification relies on a large number of labeled samples. However, there are only a few training samples for some sub-categories in many fields, e.g. rare cases in medical research [5], unbalanced time series classification [6], and newly species in biological research [7]. Because of the obvious technical merits, many few-shot learning (FSL) methods are proposed to classify the object images with a few training samples in recent years [8], [9], [10]. Nevertheless, they cannot capture the subtle features in fine-grained classification tasks. Therefore, solving the few-shot fine-grained (FSFG) image classification task has attracted intensive attention from the research community. The FSFG learning domain, however, remains ripe for improvement, chiefly due to two persistent challenges:

1) The intra-class distances of sub-categories may be very large. The appearance and background are two important factors that affect the inter-class distance, which play a negative role in the FSFG task. As shown in Fig. 1, even in the same row, the posture and background are very different. 2) the inter-class distances among different sub-categories may be small. Note

Fig. 1. Illustration of a “5-way 5-shot” FSFG task. Different columns represent different bird species: “Black-footed Albatross”, “Pelagic Cormorant”, “Herring Gull”, “Sooty Albatross”, “California Gull”. Different rows represent different poses, “still” and “fly”, and backgrounds, “ground”, “sky”, “grass”, and “sea”.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 18, 2024 at 13:10:26 UTC from IEEE Xplore. Restrictions apply.
that only local subtle features can be used to distinguish different sub-categories. For different tasks, the features used for classification may be very different. For example, in each column of Fig. 1, only the beak or wings of birds are different, so it’s best to pay more attention to the beak or wings. Moreover, in the last query image of the query set, because of the angle of view and distance from the camera, there is only the head of the bird, so it’s best to focus on the bird’s head for both query and support images. These factors collectively heighten the classification challenge, emphasizing the need for nuanced discrimination among subtly varying sub-categories.

For the first problem, we employ the complex network structure and training strategy to obtain a robust feature of each sub-category. Some methods based on feature enhancement [11], [12], [13] were proposed to improve it. For example, Wei et al. [11] augmented a single coding feature with a self-bilinear encoding network. Huang et al. [13] proposed a pairwise bilinear pooling network to extract second-order descriptors for obtaining more precise feature representations. However, they regard the image as a whole space in terms of pixel or feature to enhance features, which ignore the structure of the target object and cannot avoid the negative impact of the background. For the second problem, Zhu et al. [14] proposed a multi-attention method to get the discriminative parts of images. Huang et al. [15] employed a new low-rank bilinear pool operation to capture subtle differences between the support set and query set. However, they do not consider that the features used to distinguish sub-categories may vary significantly for different tasks, and the samples in the support set and query set may also respond to different parts of the target object. Therefore, it not only needs to obtain more adaptable and comprehensive features of the target object but also align the features in the support set and query set.

In this work, we propose a novel few-shot fine-grained image classification network (FicNet) with the multi-frequency neighborhood (MFN) and double-cross modulation (DCM). MFN first captures smooth spatial structure based on important relationships among the neighborhoods and self-similarity, which can reduce the change sensitivity of intra-class appearance. Then it decomposes the structural features into multi-frequency components, isolating compact structural patterns and eliminating irrelevant features. DCM module consists of the bi-crisscross (BCC) and the double 3D cross-attention (DCA). BCC is used to extract more context information from different regions, which modulates the embedding to hold more complete target information. DCA refines the correlation between the support and query samples to generate a reliable co-attention map, which modulates the embedding to contain the task-level discriminative features. Then the samples can respond to the same parts of the target object. Extensive experiments are performed to show that FicNet significantly outperforms existing methods on fine-grained benchmark datasets. Our technical contributions are summarized as follows:

- Proposing a novel and compact end-to-end network, which adeptly addresses FSFG classification through targeted capture and modulation strategies.
- Designing module MFN to fuse spatial and frequency information, which can capture multi-frequency structural representation to reduce the influence of appearance and background on intra-class distance.
- Constructing Implementing module DCM to modulate the representation from both context and task-specific information, which makes the involved samples respond to the same parts and identify the subtle inter-class differences accurately.

The rest of the paper is organized as follows. First, the related work on fine-grained image classification and associate learning methods is briefly reviewed in Section II. Then, the adopted methodology of FicNet is presented in Section III. Section IV presents the detailed experimental settings and results to verify the effectiveness of FicNet. Furthermore, Section V provides extensive ablation studies on every basic component of FicNet. The last section concludes the paper.

II. THE RELATED WORK

To demonstrate the development background of FicNet, we briefly review the related work on few-shot learning, fine-grained image classification, few-shot fine-grained learning, and frequency domain learning.

- **Few-shot learning**: FSL simulates the human ability to acquire knowledge from a few samples through generalization and analogy. In recent years, many metric learning-based FSL methods have emerged. For example, Snell et al. [16] proposed a simple and efficient network ProtoNet for few-shot learning, which performed classification by calculating Euclidean distance from the prototype expression of each category. In [17], Zhang et al. employed second-order relational descriptors from feature maps to capture the similarity between the support-query pairs. Finn et al. [18] provided the most typical meta-learning method named model-agnostic meta-learning algorithm (MAML), which can learn a set of good initialization parameters. Furthermore, Wang et al. [19] devised a graph complemented latent representation (GCLR) network to learn a better representation. To alleviate the over-fitting phenomenon in FSL from the estimation of data distribution, Yang et al. [20] used calibrated distribution to provide more diverse generations using statistical information from classes with abundant samples.

- **Fine-grained image classification**: For many years, fine-grained image classification has been a hot topic in the field of computer vision. Deep fine-grained classification methods can be broadly divided into two categories: regional features-based methods and global features-based methods. Zhang et al. first combined R-CNN into a fine-grained classifier with geometric prior and used object detection to locate local regions [21]. For easy deployment, Zhang et al. [22] provided a fine-grained image categorization system, which only classified labels for training images. Different from simply locating different parts, the approach given in [23] attempted to mine complementary information from multiple granularity parts. The method designed in [24] can not only capture distinguishing parts but also use knowledge in text description for interactive alignment.
In contrast, the global features-based methods extract features from the whole image without explicit localization of the target. As a typical work, Feng [25] used kernel approximation to introduce high-order information and then adopted statistical information to enhance modeling ability.

- **Few-shot fine-grained learning:** Wei et al. [11] proposed the first FSFG model, which used a self-bilinear encoder to capture subtle image features. Li et al. [12] further replaced the naive self-bilinear encoder with covariance pooling and designed a covariance metric network, which measured the relationship between a query sample and each category by the distribution consistency. Different from these self-linear methods, Huang et al. [13] provided a novel pairwise-bilinear pooling method to compare subtle differences between fine-grained categories. Then they further employed a new low-rank bilinear pool operation [15], and designed a feature alignment layer to match the support features with the query features. Meanwhile, Wang et al. [26] proposed a data enhancement method FOT (Foreground Object Transformation), which used the posture transformation generator to generate additional samples of new sub-categories. Zhu et al. [14] provided a multi-attention meta-learning method that exploited the attention mechanism of the base learner and task learner to capture the different parts of an image. Moreover, Ruan et al. [27] explored a spatial attentive comparison network (SACN), which consists of three separate modules for feature extraction, selective comparison of similarity, and classification.

- **Frequency domain learning:** Ehrlich and Davis [28] creatively proposed a model-conversion algorithm to convert the CNN model from the spatial domain to the frequency domain. Furthermore, Wang et al. [29] directly exploited the frequency modality as complementary information and capture the local contextual incoherence to enhance boundary consistency. To avoid such a complex model transition procedure, Xu et al. [30] analyzed the spectral bias and provided a frequency selection method to identify the trivial frequency components. Meanwhile, Qin et al. [31] considered the channel representation problem as a compression process using frequency analysis. They proved that the conventional global average pooling is a special case of feature decomposition in the frequency domain.

Inspired by the related works, FicNet creatively addresses the FSFG challenge through the integration of MFN and DCM, setting it apart from conventional image classification techniques. First, MFN captures rich and compact structural patterns to reduce the impact of background variation on intra-class distance by fusing spatial and frequency information. Then, DCM modulates the embedding process to contain more complete class-related details based on global context and task-specific information. It can accurately identify the subtle inter-class differences.

### III. The Adopted Methodology

#### A. Problem Definition

An FSFG problem is usually formalized as a “$N$-way $K$-shot” task, and the dataset is divided into training set $D_{tr} = \{(I_i, y_i), y_i \in C_i\}_{i=1}^{N_{tr}}$ and testing set $D_{test} = \{(I_j, y_j), y_j \in C_j\}_{j=1}^{N_{test}}$, where $I_i$ is the $i$-th fine-grained sample in $D_{tr}$ and $y_i$ is its class label, $I_j$ is the $j$-th fine-grained sample in $D_{test}$ and $y_j$ is its class label; the label sets satisfy $C_i \cap C_j = \emptyset$, $N_{test}$ is the number of categories in test set. We organize experiments in an episodic meta-training setting. Considering that both $D_{tr}$ and $D_{test}$ consist of many episodes, we randomly select $N$ different categories in each episode and randomly choose $K$ and $P$ images in each category. They constitute support set $S = \{(I_i, y_i)\}_{s=1}^{N_K}$ and query set $Q = \{(I_q, y_q)\}_{q=1}^{N_P}$, respectively. Every episode is iteratively sampled from $D_{tr}$ to train FicNet.

#### B. Motivation

The main difficulties of FSFG classification lie in the small number of samples, large intra-class distance, and small inter-class distance. Accordingly, there are two solutions: few-shot learning [12], [32] and reducing feature dimensions to prevent overfitting caused by few samples [15]. The former can obtain more precise features by changing the measurement method or using attention to enhance the representation ability of metric space. So, it can achieve more concise classification and better portability.

To address the large intra-class distance in FSFG classification, two models proposed in [10], [33] use complex network structures and training strategies to obtain robust features for each class, but the few sample problem make these models less effective. Most importantly, these methods treat images as a whole space without considering that large intra-class distance is often caused by irrelevant backgrounds. Therefore, we consider capturing the features that are anti-interference to the irrelevant background from the spatial structure. It can reduce the intra-class distance while ensuring a small number of training parameters.

To address small inter-class distance, two models proposed in [27], [34] measure the relationship between query samples and each category with novel measurements or multilayer attention. However, they do not realize that the characteristics used to distinguish categories may vary greatly depending on the task, and samples in support sets and query sets may respond to different parts of the target object. So the methods are inefficient.

In contrast, we modulate features containing more complete features of the target object and align features in the support set and query set while keeping the efficiency as high as possible. In addition, how to better balance the two challenges is overlooked. Therefore, we consider solving the two problems in a concise and coherent way and ensure the generalization of the model.

Distinct from the previous image classification methods in [11], [35], [36], we do not simply superimpose attention or multilevel features, or blindly require more detailed features without paying attention to task requirements. We design MFN to obtain compact structural features by fusing the spatial and frequency domain. Then we provide DCM globally and adapt to different tasks flexibly. The two modules aim to solve the two existing challenges respectively. Note that the number of parameters and computations required for model training is reduced as small as possible.

---

**Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.**
C. Architecture Overview

FicNet belongs to the metric-based methods, and the specific structure is shown in Fig. 2, which is composed of two main modules, MFN and DCM. Note that marker \( \oplus \) means element-wise sum, \( L_a \) denotes the auxiliary loss function, and \( L_c \) represents the classification loss. As for the input samples, we employ ConvNet-4 as the feature extractor to provide basic feature maps, which are fed into MFN and converted into multi-frequency structural representations by analyzing the information in the spatial and frequency domain. Then, BCC efficiently generates global context information to modulate multi-frequency structural representations, and DCA uses multiple 3D crossed convolutions to perform modulation according to the current task.

D. Multi-Frequency Neighborhood

The module MFN is employed to capture compact multi-frequency structural representations by combining the information in the spatial and frequency domain, which can reduce the change sensitivity of intra-class representations. It includes weighted neighborhood and multi-frequency components. Fig. 3 illustrates the overall structure of MFN, where \( \otimes \) represents the Hadamard product. The details of MFN are introduced as follows.

- **Weighted neighborhood**: Weighted neighborhood encodes local invariant structure by associating with neighborhoods of each feature in the spatial domain. Because the operations on the support image and the query image are identical, so only the operations of the query image are illustrated here.

  Given a query image \( I_q \), we employ ConvNet-4 to extract basic feature map \( F_q \in \mathbb{R}^{C \times H \times W} \). As for each feature \( x \) in \( F_q \), we extract its \( UV - 1 \) neighbors through the sliding window, where \( UV \) is the size of the sliding block. Note that the nearer neighbor points obviously have a greater impact on \( x \). So, one can obtain the different weighted neighbor representations by

  \[
  F_q(x + e) = \frac{1}{\|e\| + 1} F_q(x + e),
  \]

  where \( x \in [0, H] \times [0, W] \), \( e \in [-\frac{U+1}{2}, \frac{U-1}{2}] \times [-\frac{V+1}{2}, \frac{V-1}{2}] \). Then, one builds local spatial representation \( s \in \mathbb{R}^{C \times H \times W \times UV} \) by aggregating information of each \( x \) in \( F_q \) and its weighted neighbors \( F_q \):

  \[
  s(x, e) = \frac{F_q(x)}{\|F_q(x)\|} \otimes \frac{F_q(x + e)}{\|F_q(x + e)\|}.
  \]
Multi-frequency components: To further optimize the local spatial representation $s$, we convert it to the frequency domain and represent it with frequency components. First, the local spatial representation $s$ is reshaped into $S \in \mathbb{R}^{C \times H \times W \times UV}$, and equally divided into $M$ blocks along the channel dimension
\[ S = [S^0, S^1, \ldots, S^{M-1}], \]
where channel dimension of each part $S^m$ is $\frac{C}{M}$. Second, a two-dimensional discrete cosine transform (2D-DCT) is employed on $S^m$:
\[ Freq^m = 2\text{-D}-\text{DCT}^{m=\cdot m}(S^m) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} S^m_{h,w} D^m_{h,w}, \]
where $m \in \{0, 1, \ldots, M-1\}$, and $(i_m, j_m)$ is the index of the frequency component corresponding to $Freq^m$. In particular, the basis function $D^m_{h,w}$ is a pre-calculated constant, so this process does not introduce parameters. Moreover, the selection of index $(i_m, j_m)$ is also a key issue, and its detailed procedure is given as follows:

-Step 1: For each possible alternative frequency component corresponding to index $(m_i, m_j)$, we evaluate the performance of the model using each component separately, where $m_i \in \{0, 1, \ldots, H-1\}$, $m_j \in \{0, 1, \ldots, W-1\}$. If the frequency component can bring higher classification accuracy, it is considered to be more important.

-Step 2: Produce the top-$M$ frequency components according to the importance demonstrated in Step 1: $\{(i_1, j_1), \ldots, (i_{M-1}, j_{M-1})\}$.

Then the condensed operation integrates the scattered frequency information to get the multi-frequency feature by
\[ D = \text{Sigmoid}\left( f_c \left( [Freq^0, Freq^1, \ldots, Freq^{M-1}] \right) \right), \]
where $f_c$ is the fully connected layer.

Finally, we obtain the multi-frequency structural representation
\[ F_q' = \text{Conv}_{3,4}(D \otimes S), \]
where $\text{Conv}_{3,4}$ represents the 3D convolution for dimension reduction.

E. Double-Cross Modulation

To modulate the multi-frequency structural representation $F_q'$ more completely and make it adapt to the current task, DCM is employed to modulate the features from both context information and task-specific information. First, we use BBC to generate global context information, which is a supplement to $F_q'$. Then, DCA is used to adjust the target region according to the current task.

Bi-crisscross component: Fig. 4 depicts the detailed contents of BCC. As for the given basic feature map $F_q \in \mathbb{R}^{C \times H \times W}$ corresponding to query image $I_q$, we repeat the crisscross operation twice (represented by $*2$) on it. Compared with self-attention [37], there are three main differences. 1) Characteristics, BBC generates a sparse crisscross attention map for each position $p$ in $F_q$, which has only $H + W - 1$ weights (orange grids), while self-attention [37] generates a denser attention map and obtains local attention. 2) Information flow, the crisscross operation in BCC only gathers the context information of each position in the horizontal and vertical directions, and two same operations of sharing parameters can completely gather the information of all positions around it. While Self-attention [37] extracts a local region of pixels in each position with spatial extent $k$ centered around it, and gathers information of each position in a block. 3) Advantages, our BCC can efficiently generate global context information, which is adapted to modulate features. While Self-attention [37] verifies that attention can be a stand-alone primitive to replace convolutions for the vision model. The specific operation on $F_q$ is described as follows:

-Step 1: $F_q$ is fed into two $1 \times 1$ convolution layers to reduce channel dimension into $C''$ and a plain $1 \times 1$ convolution layer to generate three feature maps $Q$, $K$, and $V$, where $Q, K \in \mathbb{R}^{C'' \times H \times W}$ and $V \in \mathbb{R}^{C'' \times H \times W}$. Denote feature vectors of $K$ and $V$ in position $p$ as sets $\Omega_q \in \mathbb{R}^{(H+W-1) \times C''}$ and $\Phi_p \in \mathbb{R}^{(H+W-1) \times C''}$, respectively.

-Step 2: Use connection operation to extract information in the horizontal and vertical directions by
\[ R(i,p) = \text{Softmax}\left( Q_i \Omega_q^\top \right), \]
where $\text{Softmax}$ is the activation function, $Q_i$ is a feature vector of $Q$ at position $p$, $\Omega_q \in \mathbb{R}^{C''}$ is the $i$-th element of $\Omega_q$, $R(i,p)$ is a vector on $R \in \mathbb{R}^{(H+W-1) \times H \times W}$ that reveals the degree of correlation between feature $Q_p$, $\Omega_{i,p}$, $i \in \{1, 2, \ldots, H + W - 1\}$.

-Step 3: Do the gather operation to aggregate the context information of each pixel in the cross path by
\[ \text{gather}(R(i,p), \Phi_p, F_q(p)) = \sum_{i=1}^{H+W-1} R(i,p) \Phi_{i,p} + F_q(p), \]
where $\Phi_{i,p} \in \mathbb{R}^{C''}$ is the $i$-th element of $\Phi_p$, $F_q(p)$ is the feature vector of $F_q$ at position $p$.

-Step 4: Repeat the crisscross operation in Steps 1, 2, and 3 twice and obtain $F_q''$. Fig. 5 presents intuitive explanations for...
have completed the gather operations and other positions do the same. In loop two, position \( p \) can connect all other positions because the points which are vertical and horizontal to \( p \) have completed the gather operation in loop one. For example, even if position \( o \) is not in the same row or column with \( p \), they can also be connected by other positions located in the same row or column with \( p \), such as position \( s \).

Step 5: Modulate \( F_r^q \) by employing \( F_w^q \) as the complement and obtain the improved multi-frequency feature by

\[
F_q^* = F_q^w + F_q^r.
\]

Note that the operations on the support samples are the same as that on the query samples. The prototype \( F_q^* \) of support samples \( S \) can be obtained in a class-average manner.

- **Double 3D cross-attention**: To further make modulation adapt to the current task, DCA takes a pair of support prototype \( F_q^r \) and query \( F_q^w \) to produce the corresponding cross-attention maps \( A_q \) and \( A_r \), respectively. Fig. 6 shows the pipeline of DCA. First, calculate the correlation map \( A \in R^{H_1 \times W_1 \times H \times W} \) with cosine distance, where the correlation degree of position \( x_s \) in \( F_q^r \) and \( x_q \) in \( F_q^w \) can be obtained by

\[
A(x_s, x_q) = \frac{(F_q^r(x_s) \cdot F_q^w(x_q))}{\|F_q^r(x_s)\|_2 \times \|F_q^w(x_q)\|_2},
\]

where \( x_s, x_q \in [0, H] \times [0, W] \). Then use 3D double-cross convolutions to refine 4-D tensor \( A \) with the following steps:

Step 1: Reshape tensor \( A \) from \( R^{H_1 \times W_1 \times H \times W} \) into \( R^{3 \times H_1 \times W_1 \times H \times W} \), and regard the last dimension of \( A \) as the time dimension and feed it into 3D convolution.

Step 2: Perform the same operations on \( A^T \); reshape time dimensions of two results from \( HW \) into \( H \times W \) and add them to produce \( A^1 \).

Step 3: Crossly feed \( A^1 \) into two \( 3 \times 3 \times 3 \) convolutions, and integrate the results as cross-attention map \( A' \in R^{H \times W \times H \times W} \).

Step 4: Transform \( A' \) into cross-attention map \( A_q \in R^{H \times W} \) by

\[
A_q(x_q) = \frac{\exp \left( \frac{\sum_{x_r} A_q(x_r, x_q)}{|x_q|} \right)}{\sum_{x_q} \exp \left( \frac{\sum_{x_r} A_q(x_r, x_q)}{|x_q|} \right)},
\]

where \(|x_q| = HW, T \in (0, 1)\) is the temperature coefficient, \( A_q(x_q) \) represents the average matching probability between two elements in the support prototype and query sample at position \( x_q \). Similarly, cross attention map \( A_s \in R^{H \times W} \) of support prototype can be calculated by (3).

Finally, \( A_q \) and \( A_s \) are used to modulate \( F_q^* \) and \( F_s^* \), and get the final representations

\[
\begin{align*}
\hat{F}_q &= A_q F_q^*, \\
\hat{F}_s &= A_s F_s^*,
\end{align*}
\]

which are used to fuse reliable semantic-related information between them and highlight the relevant regions.

F. Learning Flow

FicNet uses metric classification for the final representation. It is an end-to-end trainable from scratch. Its loss function and training steps are introduced as follows.

- **Metric classification**: For a “N-way K-shot” task, \( N \) support category yields \( N \) different attention maps. So, one can obtain \( N \) different support representations \( \{\hat{F}(i)\}_{i=1}^N \) and the corresponding query representations \( \{\hat{F}(i)\}_{i=1}^N \) for each category. Then, compute the similarity between \( \{\hat{F}(i)\}_{i=1}^N \) and \( \{\hat{F}(i)\}_{i=1}^N \) with cosine distance, and then classify query representation into the nearest support category.

- **Loss function**: FicNet adopts a single-stage training method and combines two classification losses for end-to-end learning. As for module MFN, we classify the query feature \( \hat{F}_q = F_q + F_q' \) by adding an additional full connection layer (the layer FC is only used in the training stage), and calculate the cross entropy loss

\[
L_a = -\log \frac{\exp \left( w_c^T \hat{F}_q + b_c \right)}{\sum_{c=1}^{N_c} \exp \left( w_c^T \hat{F}_q + b_c \right)},
\]

and use it as the auxiliary loss to guide the learning of CNN-based feature extractor and module MFN, where \( N_c \) is the total number of categories in the train set \( D_r \), \( w_c \) and \( b_c \) are the weight and bias in the layer FC, respectively. On the other hand, as for the final representations \( \hat{F}_s, \hat{F}_q \) output by DCM, we build a metric classification loss \( L_c \). Considering that the FSFG problem may contain difficult negative samples, we build contrast loss

\[
L_c = -\log \frac{\exp \left( \text{dis}(\hat{F}_s^{(i)}, \hat{F}_q^{(i)}) / t \right)}{\sum_{i=1}^N \exp \left( \text{dis}(\hat{F}_s^{(i)}, \hat{F}_q^{(i)}) / t \right)}
\]

and used it as the classification loss, where \( \text{dis}(\cdot, \cdot) \) is cosine similarity, the temperature coefficient \( t \) is set as 0.2. The classification loss function guides the model to learn from the feature space, where the query samples are measured and classified correctly. Finally, the total loss function is

\[
L = L_c + \mu L_a,
\]

where \( \mu \) is a hyper-parameter that balances the loss term.
Fig. 6. The structure of DCA, where $\oplus$ represents sum after Softmax, $\odot$ represents transpose.

Algorithm 1: Meta-Training Phase.

| Input | $D_{tr} = \{(I_i, y_i), y_i \in C^*\}_{i=1}^{N_{val}}$; learning rate $\alpha$; hyper-parameter $\mu$ |
|------|--------------------------------------------------|
| Output: | Trained model $\varphi \subseteq \{\varphi_{\text{cnn}}, \varphi_{\text{mfn}}, \varphi_{\text{dcn}}\}$ |
| Randomly initialize model parameters $\varphi_{\text{cnn}}, \varphi_{\text{mfn}}, \varphi_{\text{dcn}}$. Initialize loss $L = 0$; |
| while not done do | |
| for all samples in $T_i$ do | |
| $F_{s_{i}}, F_{q_{i}} \leftarrow \varphi_{\text{cnn}}(I_{s_{i}}, I_{q_{i}})$; |
| $F'_{s_{i}}, F'_{q_{i}} \leftarrow \varphi_{\text{mfn}}(F_{s_{i}}, F_{q_{i}})$; |
| $L_{a_{i}}(\varphi) \leftarrow \frac{1}{N_{P}} \sum_{(x_{P}, y_{P}) \in P} L_{a}(F_{s_{i}}, y_{P})$; |
| $F_{s_{i}}, F_{q_{i}} \leftarrow \varphi_{\text{dcn}}(F_{s_{i}}, F'_{s_{i}}, F_{q_{i}}, F'_{q_{i}})$; |
| $L_{c_{i}}(\varphi) \leftarrow \frac{1}{N_{P}} \sum_{(x_{P}, y_{P}) \in P} L_{c}(F_{s_{i}}, y_{P})$; |
| $L_{T_{i}}(\varphi) \leftarrow L_{c_{i}}(\varphi) + L_{a_{i}}(\varphi)$; |
| end | |
| $\varphi \leftarrow \varphi - \alpha \frac{1}{|T|} \sum_{T_i} L_{T_{i}}(\varphi)$; |
| end | |

- **Training procedure:** The pseudocodes for implementing the meta-training process are given in Algorithm 1.

IV. PERFORMANCE EVALUATION

Four benchmark datasets and the experimental details are first introduced to support the following performance evaluation of FicNet over some state-of-the-art methods.\(^1\)

A. Datasets

FicNet is evaluated on three fine-grained benchmark datasets: “CUB-200” (200 kinds of birds, 11,788 images) [38], “Stanford Dogs” (120 kinds of dogs, 20,580 images) [39], “Stanford Cars” (196 kinds of cars, 16,185 images) [40]. In addition, we also test the generalization ability of FicNet on a general image classification dataset, mini-ImageNet [8]. It is a subset of ImageNet [41] and consists of 60,000 images distributed over 100 different categories. For fair performance comparison, we adopt the data split method used in [27], [34]. The original images are divided into two disjoint subsets: the auxiliary training set and the testing set. For each category in the auxiliary training set, we divide the data into training dataset and validation dataset. The former is used for training parameters and the latter is employed to monitor the learning process. The details of the mentioned four datasets are shown in Table I, where $N_{\text{total}}$ is the original total number of categories in the datasets; $N_{\text{val}}$ is the number of categories in the validation set.

| Dataset        | $N_{\text{total}}$ | $N_{\text{tr}}$ | $N_{\text{val}}$ | $N_{\text{test}}$ |
|----------------|--------------------|-----------------|------------------|------------------|
| CUB-200        | 200                | 120             | 30               | 50               |
| Stanford Dogs  | 120                | 70              | 20               | 20               |
| Stanford Cars  | 196                | 130             | 17               | 49               |
| mini-ImageNet  | 100                | 64              | 16               | 20               |

B. Experimental Configuration

For fair and extensive comparison, we employ ConvNet-4 and ResNet-12 as different backbones followed with the recent FSFG classification works [13], [14], [27], [34]. All the samples used for training and testing are resized to $84 \times 84$. We set $U = V = 5$ for module MFN. As for module DCM, considering memory and time, the size of 3D convolutions is set as $3 \times 3 \times 3$, $T$ of DCA is set as 1 for “Stanford Dogs” and 2 for others; parameter $\mu$ in (5) is set as 0.7; SGD is used to optimize the parameters and the learning rate $\alpha$ is initially set as 0.1. We use the episodic training paradigm to train FicNet. For the “$N$-way $K$-shot” task, we randomly select $N$ categories, and get $K$ arbitrary samples from each category as the support set, and then randomly choose 15 samples as the query set for evaluation. Finally, we report top-1 average classification accuracy with 95% confidence intervals of 1,200 test episodes.

\(^1\)The source code of this paper and the test images used are publicly available at https://github.com/ChengqingLi/FicNet.
TABLE II
PERFORMANCE COMPARISON OF FicNet WITH TYPICAL FSFG IMAGE CLASSIFICATION METHODS OVER THREE DATASETS (%)  

| Method          | Backbone       | CUB-200 5-way 1-shot | CUB-200 5-way 5-shot | Stanford Dogs 5-way 1-shot | Stanford Cars 5-way 1-shot |
|-----------------|----------------|----------------------|----------------------|-----------------------------|-----------------------------|
| Matching Net [8] |                | 45.30 ± 1.03         | 59.50 ± 1.01         | 35.80 ± 0.99                | 47.50 ± 1.03                |
| ProtoNet [16], [42] |               | 37.36 ± 1.00         | 45.28 ± 1.03         | 37.59 ± 1.00                | 48.19 ± 1.03                |
| MAML [18]       |                | 54.92 ± 0.95         | 73.18 ± 0.77         | 44.64 ± 0.98                | 60.20 ± 0.80                |
| RelationNet [43] |                | 59.58 ± 0.94         | 77.62 ± 0.67         | 43.05 ± 0.86                | 63.42 ± 0.76                |
| PCM [11]        |                | 42.10 ± 1.96         | 62.48 ± 1.21         | 28.78 ± 2.33                | 46.92 ± 2.00                |
| GNN [44]        |                | 51.83 ± 0.98         | 63.69 ± 0.94         | 46.31 ± 0.98                | 62.27 ± 0.75                |
| CovNet [12]     |                | 52.42 ± 0.76         | 63.76 ± 0.64         | 49.10 ± 0.76                | 63.04 ± 0.65                |
| DNN [35]        |                | 46.84 ± 0.81         | 74.92 ± 0.64         | 45.41 ± 0.76                | 63.51 ± 0.62                |
| PARNN [13]      | ConvNet-4      | 63.36 ± 0.88         | 74.71 ± 0.60         | 45.65 ± 0.71                | 61.24 ± 0.62                |
| LRPARNN [15]    |                | 63.63 ± 0.77         | 76.06 ± 0.58         | 45.72 ± 0.75                | 60.94 ± 0.66                |
| ATL [36]        |                | 60.91 ± 0.91         | 77.05 ± 0.67         | 54.49 ± 0.92                | 73.20 ± 0.69                |
| BSNet (R&C) [34]|                | 65.89 ± 1.00         | 80.99 ± 0.63         | 51.06 ± 0.94                | 68.60 ± 0.73                |
| FOT [26]        |                | 67.46 ± 0.68         | 83.19 ± 0.43         | 49.32 ± 0.74                | 68.18 ± 0.69                |
| TOAN [45]       |                | 65.34 ± 0.75         | 80.43 ± 0.60         | 49.30 ± 0.77                | 67.17 ± 0.49                |
| SACN [27]       |                | 71.50 ± 1.62         | 79.77 ± 0.96         | 64.30 ± 1.74                | 71.65 ± 1.00                |
| DLG [42]        |                | 64.77 ± 0.90         | 83.31 ± 0.55         | 47.77 ± 0.86                | 67.07 ± 0.72                |
| FRN+TDM [10]    |                | 70.79 ± 0.24         | 80.03 ± 0.14         | 57.64 ± 0.22                | 75.77 ± 0.16                |
| NPDNet [32]     |                | 64.74 ± 0.90         | 80.52 ± 0.63         | 56.21 ± 0.86                | 74.82 ± 0.84                |
| BSNet (R&C) [34]| ResNet-10      | 69.73 ± 0.97         | 82.85 ± 0.61         | 58.83 ± 0.85                | 76.60 ± 0.69                |
| MML (KL) [46]   | ResNet-12      | 69.21 ± 0.75         | 82.85 ± 0.61         | 59.05 ± 0.23                | 75.79 ± 0.21                |
| Meta DeepBDC [47]| ResNet-12      | 83.60 ± 0.40         | 92.86 ± 0.21         | 66.97 ± 0.99                | 84.07 ± 0.66                |
| FOT [26]        | ResNet-18      | 72.56 ± 0.65         | 87.22 ± 0.35         | 58.03 ± 0.27                | 74.69 ± 0.25                |
| FicNet Trend    | ConvNet-4      | 75.27 ± 0.61         | 88.48 ± 0.37         | 64.74 ± 0.69                | 79.23 ± 0.46                |
| FicNet Trend    | ResNet-12      | 80.97 ± 0.57         | 93.17 ± 0.32         | 72.41 ± 0.64                | 85.11 ± 0.37                |

\(\text{\#NDPNet uses a novel feature re-abstraction network as the extractor.}\)
\(\text{\#Neither experiment results on the specified settings nor open source codes are available for the corresponding method, so accurate comparison results are unavailable.}\)

The best results are marked in bold.

C. Comparison With the State-of-the-Art Methods

To evaluate the effectiveness of FicNet on fine-grained datasets, we compare it with some state-of-the-art methods, including several specific FSFG methods and four typical FSL methods. Table II lists the comparison results on three fine-grained datasets including “CUB-200”, “Stanford Dogs” and “Stanford Cars” with standard “5-way 1-shot” and “5-way 5-shot” tasks. Just referenced the previous counterparts, the best results are marked in bold, and the second-best results with ConvNet-4 and ResNet are marked in wavy underline and dashed underline, respectively. Especially, the numbers beside symbol ↑ indicate an improvement percentage of FicNet with respect to the second best result.

From Table II, one can see that when ConvNet-4 is used as the feature extractor, FicNet can achieve improvement of 3.77%, 0.44% and 8.95% over the second best method on the three datasets for the “5-way 1-shot” tasks, respectively. As for the “5-way 5-shot” task, FicNet achieves accuracy improvement of 0.45%, 3.46% and 0.31% over the second best method on the three datasets, respectively. When Resnet-12 is used as the feature extractor, in general, FicNet also significantly outperforms the other comparable methods. It can obtain an average accuracy of 80.97%, 72.41%, and 86.81% on the three datasets for the “5-way 1-shot” task, respectively. As for the above task, the corresponding accuracy ratios are 93.17%, 85.11%, and 95.36%. Due to the limitation of presentation space, only some typical results are given in Table II. Moreover, the effectiveness of FicNet is unrelated to the selected backbone, which verifies its robustness and compatibility.

D. Generalization Ability of ResNet Over the General FSL Tasks

Most recent image classification methods [48], [49], [50] use deeper convolutional neural networks as feature extraction backbones, such as the ResNet family. To illustrate the generalization ability of FicNet, we perform experiments on mini-ImageNet. Here, we select ResNet-12 as the backbone to perform the comparative experiment on the “5-way 1-shot” and “5-way 5-shot” tasks. Table III shows the comparison results, where the best results are marked in bold, and the numbers beside symbol ↑ indicate improvement of FicNet over the baseline (ProtoNet*). As can be seen from Table III, even compared with those methods whose backbone network is ResNet-18, ResNet-34, or WRN-28-10, FicNet still demonstrates better performance, where “+” denotes larger backbones than ResNet-12. As for “5-way 1-shot” and “5-way 5-shot” tasks, the classification accuracy of FicNet can reach 67.82% and 82.71%, respectively. Especially, the result on “5-way 1-shot” task is superior to all the sampled randomly. The model is implemented by PyTorch on Tesla K80.
the other state-of-the-art methods, verifying that FicNet has high stability and strong generalization.

E. Visualization Results

To verify the performance of FicNet in a visual way, we perform the t-distributed stochastic neighbor embedding (t-SNE) visualization. We randomly select three “5-way 5-shot” tasks on dataset “CUB-200”, and each task includes 25 support samples and 175 query samples. The feature distributions of modules MFN and DCM are plotted in Fig. 7, where each row represents a “5-way 5-shot” task, and each column represents the corresponding feature distribution.

From Fig. 7(a)–(b), one can see that the samples belonging to the same category are gradually clustered together and the intra-class distance decreases significantly, which indicates that MFN can learn discriminative class-invariant features. From Fig. 7(b)–(c), one can intuitively observe that the inter-class distance becomes larger and the classification boundary is clearer. In particular, in Fig. 7(b), some samples distributed at the edge or far from the true category center are prone to be misclassified. After using DCM to do modulation based on global context and task-specific information, the misclassified samples are pulled back to the correct position, which illustrates that DCM can obtain complete and adaptive features that are conducive to good sample matching.

F. Effect of Different Metric Functions

As we all know, the metric function may be a key factor for metric-based learning methods. Therefore, we construct experiments to further explore the performance of FicNet with different metric functions such as Manhattan distance, Euclidean distance, and Cosine similarity distance. The results are reported in Table IV, showing that FicNet obtains 80.97% and 93.17% on “1-shot” and “5-shot” tasks with cosine similarity metric function, which is the highest accuracy among all comparable metric functions. Moreover, the lowest accuracy can still achieve 80.29% and 92.39% on “1-shot” and “5-shot” tasks, which illustrates different metric functions can achieve similar accuracy. So, the influence of different metric functions is acceptable.

V. ABLATION STUDY

To illustrate the effects of different modules of FicNet, we provide extensive ablation studies using ResNet-12 as the backbone.

A. Effectiveness Analysis of Different Modules

To evaluate the effectiveness of modules MFN and DCM, we conduct experiments with “5-way 1-shot” and “5-way 5-shot” tasks on dataset “CUB-200”, where ProtoNet* is used as the baseline. Table V shows the specific experimental results including accuracy and the number of additional parameters, where symbols “+” and “−” indicate the corresponding module is employed and not, respectively.

As can be observed from Table V, compared with the baseline, MFN achieves 5.78% and 5.84% accuracy improvements on “1-shot” and “5-shot” tasks, respectively. In contrast, DCM can achieve 5.54% and 6.23% improvements on the two tasks, respectively. They also verify the effectiveness of MFN and DCM. Moreover, combining modules MFN and DCM can achieve classification accuracy of 80.97% and 93.17% for the two tasks, respectively. All these results verify that the two modules are orthogonal and complementary, and MFN and DCA are both effective for solving the FSFG problem.
Fig. 7. t-SNE visualization results of FicNet: (a) distribution of basic features; (b) distribution of multi-frequency structural features; and (c) distribution of final features.

TABLE VI
THE ACCURACY ANALYSIS OF THE WEIGHTED NEIGHBORHOOD

| Weighted neighborhood | 5-way accuracy (%) | 1-shot | 5-shot |
|-----------------------|--------------------|--------|--------|
| w/o                   | 79.63              | 92.18  |
| w/                    | 80.97              | 93.17  |

B. The Performance Analysis of MFN

To further illustrate the effectiveness and generalization of MFN, we provide experiments on the frequency component and its number in MFN to analyze the performance with other methods.

- **Influence of the weighted neighborhood**: To illustrate that the weighted neighborhood is reasonable and significant for MFN in the spatial domain, we perform corresponding experiments on “CUB-200”. As shown in Table VI, whether it is “5-way 1-shot” or “5-way 5-shot” task, the model performance can be improved to a certain extent after using the weighted neighborhood to encode local invariant structure, the improvements are 1.34% and 0.99%, respectively. The results verify the necessity and effectiveness of the weighted neighborhood.

- **Influence of individual frequency components**: To study the impact of different frequency components on the performance of MFN, we use only one frequency component of FicNet at a time to evaluate the separate classification accuracy. We divide the 2-D DCT frequency space into $5 \times 5$ blocks and construct 25 experiments on mini-ImageNet with the “5-way 1-shot” task. The experimental results are shown in Fig. 8 in the form of a $5 \times 5$ matrix. As can be seen from Fig. 8, the lower frequency components generally have better performance. Their accuracies are different no matter whether the frequency components are close or far, which shows that different frequency components contain different information, and dispels the doubt that the use of multiple frequency components brings information redundancy. Finally, among all frequency components, the difference scope in classification accuracy is within 2%. Therefore, even if multiple frequency components are not selected in a targeted manner, it can bring a certain degree of performance improvement, which illustrates that it is an effective innovation to combine the information in the spatial and frequency domain to encode the features.
Influence of the number of frequency components: To further illustrate the influence of the number of frequency components $M$ on model accuracy, we choose $M = 1, 2, 4, 8, 12, 16, 25$ to do experiments on dataset “CUB-200”. Fig. 9 demonstrates that the classification accuracy increases obviously when increasing the number of multiple frequency components from 2 to 12 in the “5-way 1-shot” and “5-way 5-shot” tasks. This is attributed to that different frequency components contain different information, and more information can be extracted from the frequency domain. Moreover, two curves reach peak values of 80.97% and 93.17% at $M = 12$, respectively. This also means that 12 frequency components are sufficient to capture useful features. Therefore, $M$ is set as 12 in FicNet.

Effect of MFN in capturing structure representation: To verify the ability of MFN in capturing the structure representation, we show some test images in “CUB-200” dataset and the corresponding class activation maps in Fig. 10. As shown in Fig. 10(c), compared with the activation map of the basic CNN feature extractor in Fig. 10(b), the structure of the target object can be detected after using weighted neighborhood. In Fig. 10(d), when using MFN combined weighted neighborhood with the frequency information, the structure and the contour are more compact and clearer, while unrelated information to the structure such as background interference is weakened. It shows that module MFN has good ability to capture compact and complete structure representation.

Boosting other classification methods by MFN: We explore the generalization and applicability of MFN by combining it with other models. Here, we combine MFN with ProtoNet* in [16] and CAN in [49] to illustrate the performance. Table VII lists the classification accuracy of datasets “CUB-200” and mini-ImageNet with the “5-way 1-shot” task. As can be seen from Table VII, MFN improves the accuracy of ProtoNet* by 5.78% on dataset “CUB-200” and 4.43% on dataset mini-ImageNet. It also obtains 1.96% and 3.08% improvement over the results obtained with CAN on the two datasets, respectively. All these results convince us that MFN can boost the performance of other methods significantly.

Comparing MFN with other similarity-based modules: To further show the effectiveness of MFN, we compare it with other similarity-based modules by adding them to the baseline model ProtoNet*. The module SCE in [61] calculates the cosine similarity between the referenced position and its adjacent ones, and connects their similarity in a fixed order of channel dimension. On the contrary, MFN does not limit the observation of different neighboring relations. SCR in [54] uses self-similarity to learn the relation pattern of “how each feature is related to its neighbors”. However, it does not consider that different neighbors have different importance. MFN uses weighted neighbors and integrates

| method       | CUB-200 | mini-ImageNet |
|--------------|---------|---------------|
| ProtoNet* [16] | 73.13   | 62.39         |
| ProtoNet*+MFN | 78.91   | 66.82         |
| CAN [49]     | 77.22   | 63.85         |
| CAN+MFN      | 79.18   | 66.93         |

Fig. 8. Classification accuracy (%) on mini-ImageNet using different frequency components.

Fig. 9. Classification accuracy of FicNet using a different number of frequency components.

Fig. 10. The capturing performance of structure representation through various class activation maps: (a) origin images; (b) basic CNN feature extractor; (c) weighted neighborhood; and (d) MFN.
frequency information to capture more compact structural patterns. As shown in Table VIII, compared with SCE, the improvement of the MFN is 3.63% on mini-ImageNet, and it outperforms all the other similarity-based modules. The results verify the effectiveness of MFN.

C. The Performance Analysis of DCM

To deeply illustrate the performance of DCM, we study how to select the number of crisscross operations of BCC and analyze the performance of BCC and DCA for the modulation.

- **Influence of the number of crisscross operations:** To show the effect of using different numbers of crisscross operations in BCC, Fig. 11 depicts the variation curve of classification accuracy, memory, and time with the number L. As shown in Fig. 11, after adding one crisscross operation into FicNet, the performances of “1-shot” and “5-shot” tasks are improved by 1.11% and 1.16%, respectively. These can effectively demonstrate the importance of supplementing context information. Then, if increase number L from 1 to 2, that is, using BCC component, the model can improve the performance under both tasks by another 0.41%. Furthermore, in Fig. 11, we also observe that the memory occupancy and computation time continue to increase with respect to L. Therefore, to balance the performance and computational resources, L = 2 is optimal for FicNet.

- **Effectiveness analysis of BCC and DCA of DCM:** To study the effect of BCC and DCA of DCM, we employ ablation experiments on dataset “CUB-200” for “1-shot” and “5-shot” tasks. Note that if the module is employed to FicNet, it is represented with “+” and “−” otherwise. The results are illustrated in Table IX, which show that BCC achieves 0.27% and 0.15% improvements over the baseline for “1-shot” and “5-shot” tasks. In contrast, DCA obtains 1.02% and 0.45% improvements on the two tasks, respectively. Moreover, combining BCC with DCA can achieve a classification accuracy of 2.06% and 2.3% improvements over the baseline for the two tasks, respectively. All these verify that the combination of BCC and DCA can further improve the classification accuracy and bring benefits to the model, which also shows that it is necessary to integrate the global context information into the embedding process.

- **Comparison of DCA with other cross-attention modules:** To further analyze the performance of the DCA component, we construct experiments with DCA and other cross-attention modules under the same experimental setting. Similar to the DCA component, CAN in [49] and the cross-correlational attention (CCA) in RENet [54] also focus on task-specific information by using cross attention, and the former directly averages 4D cross-correlation between images into 2D tensor, the latter learns reliable co-attention via 4D convolutional filter. Compared with them, DCA uses 3D double-cross convolutions, which crosswise and progressively refine the 4D cross-correlation tensor while maintaining the details of geometric space. The DCM module in FicNet not only contains the DCA component but also focuses on context information with the BCC component. The comparison results are shown in Table X. As shown in the first three rows of Table X, the DCA component performs best. Especially when replacing CAN and CCA with DCA components, the classification accuracy is improved by 0.71%, 0.34% and 1.18%, 0.74% on “CUB-200” and mini-ImageNet, respectively. The results illustrate that DCA can avoid the collapse of key details caused by CAN, and obtain more precise task-level attention than CCA. In addition, when replacing CCA with

![Fig. 11. Experimental results of FicNet using the different number of criss-cross operations.](image-url)
the DCM module, the classification accuracy is improved by 1.08% and 1.97% on “CUB-200” and mini-ImageNet respectively. And the last two rows of Table X demonstrate that the classification accuracy can be improved after adding BCC components. The results also explain that context and task information can both supplement each other in DCM.

VI. DISCUSSION

As shown by the results listed in the above tables and figures, FicNet is superior to the existing FSFG image classification methods. We discuss the underlying reasons and possible extensions as follows.

- Unlike the models that only focus on extracting embedding [33], extracting local information [36], [50] and enhancing data [26], [35], we propose a unique FSFG method based on the principles of tight coupling and complementary, which explores the structure of the object itself, the global context, and the interactive task information between the support image and query image. It can narrow the intra-class distance, and make the boundaries between inter-class clearer, demonstrating strong generalization ability on mini-ImageNet.

- MFN fuse spatial domain and frequency domain information at the same time. Compared with the methods in [14], [17], [46] that don’t consider using the irrelevant features, we first extracted the real target structure based on weighted neighbors, and then used multi-frequency components to obtain compact information. The experimental results in Tables II and V demonstrate the superiority of MFN, and the results in Fig. 8 are also consistent with “low frequencies are the most informative” in [30]. Moreover, MFN also can be used in other applications, such as person re-identification [62], due to large intra-class differences caused by perspective changes, occlusion, and background noise. So, we suggest a module similar to MFN that starts from self-similarity and internal structure to identify human bones.

- DCM is used for modulation based on context and task information. Compared with the methods [33], [35], [44] that blindly pursued details in non-targeted areas, DCM employed the completeness of the target object and performed modulation based on the task. The effect of these operations is shown in Table II. Moreover, DCM also can be employed in other applications, such as facial expression recognition [63], facial expressions are expressed through many facial regions, but their inter-class differences are subtle and local. Therefore, we suggest trying a module similar to DCM that takes into account both global and subtle local representations.

VII. CONCLUSION

In this paper, we designed a high-accuracy few-shot fine-grained image classification method by capturing and modulating with MFN and DCM. MFN captured multi-frequency structural representation from spatial and frequency information, which could bring a certain degree of performance improvement. DCM modulated features from both global context and task-specific information. Moreover, The proposed FicNet did not require bounding box or part annotation in the new representation for metric classification. Extensive experiments illustrated that FicNet outperformed the previous state-of-the-art counterparts on some fine-grained datasets. However, its performance may be less significant when dealing with richer datasets. Extending its generalization and effectiveness to efficiently deal with more complex scenes and computer vision tasks, such as salient object detection, deserves further exploring.

REFERENCES

[1] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, “Multi-channel deep networks for block-based image compressive sensing,” IEEE Trans. Multimedia, vol. 23, pp. 2627–2640, 2021.

[2] S. Zhou, X. Deng, C. Li, Y. Liu, and H. Jiang, “Recognition-oriented image compressive sensing with deep learning,” IEEE Trans. Multimedia, vol. 25, pp. 2022–2032, 2023.

[3] G. Van Horn et al., “Benchmarking representation learning for natural world image collections,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12875–12888.

[4] L. Guan, L. Jia, Z. Xie, and C. Yin, “A lightweight framework for obstacle detection in the railway image based on fast region proposal and improved YOLO-tiny network,” IEEE Trans. Instrum. Meas., vol. 71, 2022, Art. no. 5009116.

[5] R. Feng et al., “Interactive few-shot learning: Limited supervision, better medical image segmentation,” IEEE Trans. Med. Imag., vol. 40, no. 10, pp. 2575–2588, Oct. 2021.

[6] H. Zhu et al., “A driving behavior risk classification framework via the unbalanced time series samples,” IEEE Trans. Instrum. Meas., vol. 71, 2022, Art. no. 2503312.

[7] X. Sun et al., “Few-shot learning for domain-specific fine-grained image classification,” IEEE Trans. Ind. Electron., vol. 68, no. 4, pp. 3588–3598, Apr. 2021.

[8] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3637–3645.

[9] H. Chen, H. Li, Y. Li, and C. Chen, “Multi-scale adaptive task attention network for few-shot learning,” in Proc. IEEE 26th Int. Conf. Pattern Recognit., 2022, pp. 4765–4771.

[10] S. Lee, W. Moon, and J.-P. Heo, “Task discrepancy maximization for fine-grained few-shot classification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 5331–5340.

[11] X.-S. Wei, P. Wang, L. Liu, C. Shen, and J. Wu, “Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples,” IEEE Trans. Image Process., vol. 28, no. 12, pp. 6116–6125, Dec. 2019.

[12] W. Li et al., “Distribution consistency based covariance metric networks for few-shot learning,” in Proc. AAAI Conf. Artif. Intell., 2019, pp. 8642–8649.

[13] H. Huang, J. Zhang, J. Zhang, Q. Wu, and J. Xu, “Compare more nuanced: Pairwise alignment bilinear network for few-shot fine-grained learning,” in Proc. IEEE Int. Conf. Multimedia Expo, 2019, pp. 91–96.

[14] Y. Zhu, C. Liu, and S. Jiang, “Multi-attention meta learning for few-shot fine-grained image recognition,” in Proc. Int. Joint Conf. Artif. Intell., 2020, pp. 1090–1096.

[15] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, “Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification,” IEEE Trans. Multimedia, vol. 23, pp. 1666–1680, 2021.

[16] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 4080–4090.

[17] H. Zhang, H. Li, and P. Koniusz, “Multi-level second-order few-shot learning,” IEEE Trans. Multimedia, vol. 25, pp. 2111–2126, 2022.

[18] C. Finn, P. Abbeel, and S. Levine, “Domain-agnostic meta-learning for fast adaptation of deep networks,” in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 1126–1135.
[19] X. Zhong, C. Gu, M. Ye, W. Huang, and C.-W. Lin, “Graph complemented latent representation for few-shot image classification,” *IEEE Trans. Multimedia*, vol. 25, pp. 1979–1990, 2023.

[20] S. Yang, L. Liu, and M. Xu, “Free lunch for few-shot learning: Distribution calibration,” in *Proc. Int. Conf. Learn. Representations*, 2021.

[21] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, “Part-based R-CNNs for fine-grained category detection,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[22] Y. Zhang et al., “Weakly supervised fine-grained categorization with part-based image representation,” *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1713–1725, Apr. 2016.

[23] L. Zhang, S. Huang, W. Liu, and D. Tao, “Learning a mixture of granularity-specific experts for fine-grained categorization,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8331–8340.

[24] K. Song, X.-S. Wei, X. Shu, R.-J. Song, and J. Li, “Bi-modal progressive mask attention for fine-grained recognition,” *IEEE Trans. Image Process.*, vol. 29, pp. 7006–7018, 2020.

[25] S. Feng, “Kernel pooling feature representation of pre-trained convolutional neural networks for leaf recognition,” *Multimedia Tools Appl.*, vol. 81, no. 3, pp. 4255–4282, 2022.

[26] C. Wang, S. Song, Q. Yang, X. Li, and G. Huang, “Fine-grained few-shot learning with foreground object transformation,” *Neurocomputing*, vol. 466, pp. 16–26, 2021.

[27] X. Ruan, G. Lin, L. Long, and S. Lu, “Few-shot fine-grained classification with spatial attentive comparison,” *Knowl.-Based Syst.*, vol. 218, 2021, Art. no. 106840.

[28] M. Ehrlich and L. S. Davis, “Deep residual learning in the JPEG transform domain,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3484–3493.

[29] J. Wang, S. Chen, Z. Wu, and Y.-G. Jiang, “FFT-TDR: Frequency-guided transformer and top-down refinement network for blind face inpainting,” *IEEE Trans. Multimedia*, vol. 25, pp. 2382–2392, 2023.

[30] K. Xu et al., “Learning in the frequency domain,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1740–1749.

[31] Z. Qin, P. Zhang, F. Wu, and X. Li, “FcaNet: Frequency channel attention networks,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 783–792.

[32] W. Zhang, X. Liu, Z. Xue, Y. Gao, and C. Sun, “NdPNet: A novel non-linear data projection network for few-shot fine-grained image classification,” 2021, arXiv:2106.06988.

[33] X. Chen and G. Wang, “Few-shot learning by integrating spatial and frequency representation,” in *Proc. IEEE 15th Conf. Robust Vis.*, 2021, pp. 49–56.

[34] X. Li et al., “BSNet: Bi-similarity network for few-shot fine-grained image classification,” *IEEE Trans. Image Process.*, vol. 30, pp. 1318–1331, 2021.

[35] W. Li et al., “Revisiting local descriptor based image-to-class measure for few-shot learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7260–7268.

[36] C. Dong, W. Li, J. Hao, Z. Gu, and Y. Gao, “Learning task-aware local representations for few-shot learning,” in *Proc. 29th Int. Conf. Int. Joint Conf. Artif. Intell.*, 2021, pp. 716–722.

[37] P. Ramachandran et al., “Stand-alone self-attention in vision models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 68–80.

[38] H.-J. Ye, H. Yu, D.-C. Zhan, and F. Sha, “Few-shot learning via embedding adaptation with set-to-set functions,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8808–8817.

[39] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, “Novel dataset for fine-grained image categorization: Stanford dogs,” in *Proc. 1st Workshop Fine-Grained Vis. Categorization IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1–2.

[40] K. Jonathan, S. Michael, D. Jia, and F.-F. Li, “3D object representations for fine-grained categorization,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 554–561.

[41] O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[42] S. Cao, W. Wang, J. Zhang, M. Zheng, and Q. Li, “A few-shot fine-grained image classification method leveraging global and local structures,” *Int. J. Mach. Learn. Cybern.*, vol. 13, pp. 2273–2281, 2022.

[43] F. Sung et al., “Learning to compare: Relation network for few-shot learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1199–1208.

[44] V. Garcia and J. Bruna, “Few-shot learning with graph neural networks,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[45] H. Huang et al., “TOAN: Target-oriented alignment network for fine-grained image categorization with few labeled samples,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 853–866, Feb. 2022.
Zhan Gao received the B.S. degree from the School of Mathematics and Information Science, Henan Normal University, Xinxiang, China, in 2020. She is currently working toward the M.S. degree with Northeastern University, Shenyang, China. Her research interests include deep learning and image classification.

Jiayi Wang received the B.S. degree in information and computation sciences in 2020 from Northeastern University, Shenyang, China, where she is currently working toward the M.S. degree in applied mathematics. Her research interests include data mining, machine learning, deep learning, and Big Data analysis.

Yange Zhou received the B.S. degree in information and computation science, Northeastern University, Shenyang, China, in 2021. She is currently working toward the M.S. degree with Northeastern University, Shenyang, China. Her research interests include deep learning and computer vision.

Chengqing Li received the M.Sc. degree in applied mathematics from Zhejiang University, Hangzhou, China in 2005 and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2008. He was a Postdoctoral Fellow with The Hong Kong Polytechnic University, Hong Kong, till 2010. From 2013 to 2014, he was with the University of Konstanz, Konstanz, Germany, under the support of the Alexander von Humboldt Foundation. He is currently the Dean with the School of Computer Science, Xiangtan University, Xiangtan, China. He has authored or coauthored more than 60 papers on the subject in the past 20 years, receiving more than 5300 citations with h-index 39. His research interests include security analysis of multimedia encryption schemes and privacy protection schemes.