Agent Attention Feature Reconstruction Network for Fine-Grained Few-Shot Image Classification

Dongfei Chang*, Jijie Wu* and Xiaoxu Li*
* Lanzhou University of Technology, Lanzhou, China
E-mail:lixiaoxu@lut.edu.cn

***Abstract*—To develop fine-grained few-shot image classification, it is crucial to study how to increase inter-class variation and reduce intra-class variation. A recently proposed method called the Bidirectional Feature Reconstruction Network (BiFRN) increases inter-class variation by using support sets to reconstruct query sets and reduces intra-class variation by query sets to reconstruct support sets. In our study, we found a problem with BiFRN: The calculation of self-attention weights is affected by noise or inaccurate information, which may cause the model to over-focus on specific locations or ignore features at other locations. To solve this problem, we proposed an Agent Attention Bidirectional Feature Reconstruction Network (AAFRN), which extends BiFRN by fusing an Agent Attention Model to generate a new feature map for each feature map. Compared with BiFRN, our method can enhance feature expressiveness, facilitate global contextual information capture within feature maps and enable weighted feature fusion across various positions. The experimental results on three widely used fine-grained image classification datasets demonstrate that the proposed method achieves competitive performance compared to other methods.**

I. Introduction

Fine-grained few-shot image classification [1] has recently become an important approach to addressing the data scarcity problem that is widely faced by fine-grained analysis [2]. According to the traditional few-shot setting, effective model transfer is achieved when limited support samples are provided. Unlike conventional methods, we deal with problems with fine-grained features, which means that the model needs to focus on learning subtle but discriminative features. This allows classification not only in terms of categories but, more importantly, can distinguish fine-grained visual differences within a class to achieve fine-grained classification between instances within the class.

In fine-grained few-shot image classification, the self-attention mechanism plays an important role [3]. This mechanism improves the classification performance and discrimination ability of the model by emphasizing the key areas in the image. Vaswani et al. [3] introduced the self-attention mechanism, leading to the development of a novel network architecture known as the Transformer. This architecture not only plays a role in natural language processing [3]–[5], but also plays a role in image classification [5]–[7]. In light of the attention mechanism, some few-shot learning methods began to leverage the self-attention mechanism. Someone proposed a Transformer-based Few-shot Embedding Adaptation (FEAT) algorithm for Few-shot learning [8]. In paper [8], the author tried to establish the mapping relationship between the sets and found that the Transformer was the best choice. The Transformer model can effectively capture the interaction between different images in the set so that the coordinated adaptation between each image can be achieved. Unlike the Transformer structure in FEAT, exclusively applied to support samples. CTX [9] employs a self-attention mechanism to compute spatial attention weights between the query sample and support sample, facilitating the learning of a category prototype aligning with the query. Subsequently, the classification of the query involves evaluating the distance from the query to the matched category prototype. The introduced few-shot classification method additionally integrates a self-attention mechanism to optimize reconstruction weights within the self-construction model and mutual reconstruction model. PVT [10] uses sparse attention patterns to reduce the computational load by decreasing the number of keys and values. While effective, these approaches sacrifice the ability to model long-range dependencies, remaining less powerful than the global self-attention mechanism. NAT [11] simulates the computation of convolution and calculates the attention of each featured neighborhood. DAT [12] constructs a deformable attention module for data-specific attention patterns. BiFormer [13] employs a two-tier routing attention mechanism to identify the region of interest for each query dynamically. Xiong et al. [14] demonstrated that the attention mechanism enables the model to capture subtle feature differences and achieve better performance in small samples. The integration of attention mechanisms in fine-grained few-shot image learning addresses data scarcity challenges, enhancing model accuracy and generalization [15]. Our research found that although BiFRN achieves the goal of increasing inter-class variation and reducing intra-class variation through the bidirectional reconstruction method, there is still room for improvement. Agent attention adopts agent tokens for aggregating and propagating global information, offering high expressiveness with low computational overhead. This method enhances feature expressiveness, facilitates global contextual information capture within feature maps, and enables weighted feature fusion across various positions.

Initially, efforts concentrated on crafting intricate network structures [1], [16], [17]. however, the outcomes fell short of expectations when juxtaposed with the original approaches. Recent trends have witnessed the surge in popularity of reconstruction-based techniques [9], [18], exemplified by approaches like few-shot Classification With Feature Map Re-

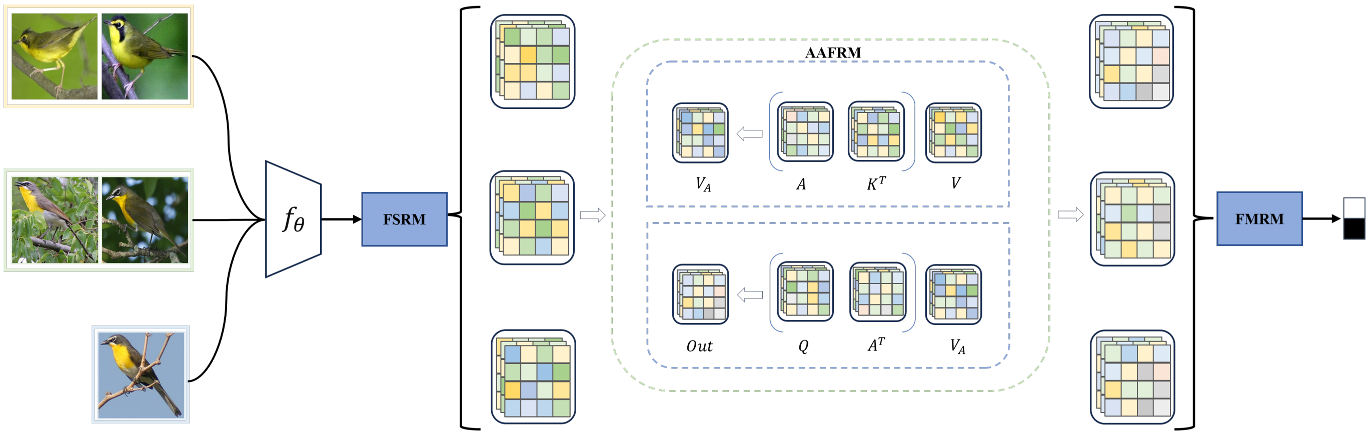

Fig. 1. The proposed agent attention bidirectional feature reconstruction network. f_θ is the feature extraction model. FSRM is a feature self-reconstruction model. AAFRM is the Agent Attention feature reconstruction model. FMRM is a feature mutual reconstruction model.

construction Networks (FRN) [19], showcasing superior performance. Through aligning support and query features, these models naturally foster fine-grained transfer. However, while these models can emphasize nuanced and distinct regions, the semantic content represented by these regions may differ across samples of the same category. This highlights the substantial intra-class discrepancies that persist within reconstruction-based methods, hindering comprehensive fine-grained learning. In essence, these approaches primarily augment inter-class disparities and fall short in mitigating intra-class variations.

Our agent attention feature reconstruction mainly consists of four Models: 1) feature extraction Model. 2) Feature Self-Reconstruction Model (FSRM). 3) Agent Attention Feature Reconstruction Model (AAFRM). 4) Feature Mutual Reconstruction Model (FMRM). With the cooperation of these modules, we can not only effectively improve the expressiveness of features and help the model to be better trained but also increase inter-class differences and reduce intra-class differences.

To summarise, our work makes a triple contributions:

- It reveals that the critical issue for fine-grained few-shot image classification is to increase inter-class variation and reduce intra-class variation.
- By integrating agent attention, the expressiveness of feature maps is improved.
- The experimental structure proves the effectiveness of the method.

II. Methodology

In this section, we will introduce the proposed method in this paper, starting with the formula of the method, followed by an overview and an in-depth description of each component.

As shown in Fig. 1, our network mainly consists of four models. The first model is the feature extraction model f_θ, which is used to extract deep convolutional image features. This part can be a residual network or a traditional convolutional network. The second model is the feature self-reconstruction Model (FSRM), which self-reconstructs the convolutional features of each image based on the self-attention mechanism. This model can make similar local features more similar and dissimilar local features more dissimilar, and it is also conducive to the mutual reconstruction of subsequent features. The third model is the Agent Attention Feature Reconstruction Model (AAFRM), which can improve the expressiveness of the model and help the model capture global context information in the feature map. The fourth Model is the Feature Mutual Reconstruction Model (FMRM), which not only reconstructs the query sample using the support sample but also reconstructs the support sample using the query sample and uses the Euclidean distance to calculate the distance between the query sample and the reconstructed query sample, as well as the distance between the support sample and the reconstructed support sample. The weighted sum of these two distances is used to classify the query sample.

A. *feature self-econstruction Model (FSRM)*

For a C-way K-shot task, we input $C \times (K + M)$ samples x_i into the feature extraction Model f_θ to extract features $\hat{x}_i = f_\theta(x_i) \in \mathbb{R}^{d \times r}$, where $r = h \times w$ and d is the channel number, h and w are the height and width of the feature. First, we shape the feature $\hat{x}_i$ as r local features in the spatial positions $[\hat{x}_i^1, \hat{x}_i^2, \cdots, \hat{x}_i^r]$, $r = h \times w$. Then the sum of the local feature $\hat{x}_i^j$ and the corresponding spatial position embedding $E_{pos} \in R^{r \times d}$ is calculated as the input of the transformer, $y_i = [\hat{x}_i^1, \hat{x}_i^2, \cdots, \hat{x}_i^r] + E_{pos}$, where E_{pos} uses sinusoidal position encoding. The output of this part is calculated according to the self-attention in Transformer Encoder. Therefore, the output $\hat{y}_i$ of this Model was calculated using (2),

To simplify we abbreviate softmax attention as (1),

$$\sigma(QK^T)V \triangleq Attn(Q, K, V) \tag{1}$$

$$\hat{y}_i = \sigma\left(y_i W_\phi^Q (y_i W_\phi^K)^T\right) y_i W_\phi^V, \hat{y}_i \in \mathbb{R}^{r \times d} \tag{2}$$

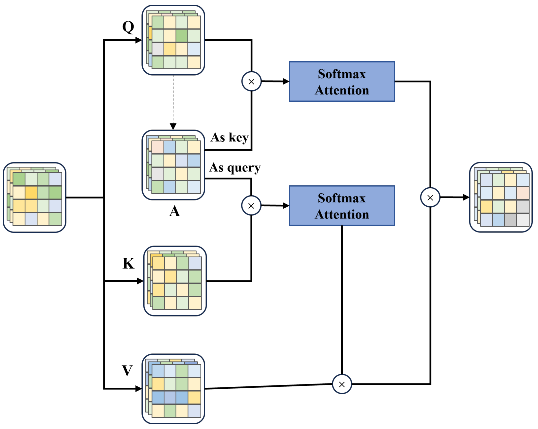

Fig. 2. Agent Attention Feature Reconstruction Model

where $\sigma(\cdot)$ represents the Softmax function, $Attn(\cdot)$ is the self-attention mechanism, $\triangleq$ indicates equivalence, W_ϕ^Q, W_ϕ^K, and W_ϕ^V are a set of learnable weight parameters with $d \times d$ size.

B. Agent Attention Feature Reconstruction Model (AAFRM)

Each feature $\hat{y}_i$ is the input of AAFRM, and the output of this Model is $\hat{z}_i \in \mathbb{R}^{d\times r}$ was calculated using (6). The Q, K, V was calculated using (3),

$$Q = yW_Q, K = yW_K, V = yW_V \tag{3}$$

where W_Q, W_K, W_V and W_A are a set of learnable weight parameters with $d \times d$ size. a represents the agent token in the agent attention, which is the result of Q adaptive average pooling was calculated using (4),

$$A = aW_A, a = Polling(Q) \tag{4}$$

where $polling(\cdot)$ is adaptive average pooling, $A \in \mathbb{R}^{d\times r}$, d is the dimension.

The agent attention consists of two softmaxes: agent aggregation and agent broadcast. Specifically, we treat A as query Q and perform an attention calculation between A, K, V to aggregate agent features V_A from all values. We call this attention calculation aggregation. Afterward, we use A as keys, V_A as value, and query matrix Q to perform a second attention calculation called agent broadcast. The final output $\hat{z}$ was calculated using (5),

$$O = \underbrace{Attn^S(Q, A, \underbrace{Attn^S(A, K, V)}_{\text{Agent Aggregation}})}_{\text{Agent Broadcast}} \tag{5}$$

It is equivalent to:

$$\hat{z} = \sigma(QA^T)\sigma(AK^T)V \tag{6}$$

where $\sigma(\cdot)$ represents the Softmax function.

C. Feature Mutual Reconstruction Model (FMRM)

The third Model is the Feature Mutual Reconstruction Model (FMRM), which contains two operations: reconstructing support features in one class given a query feature and reconstructing the query feature given support features in one class. After AAFRM, we can obtain the support features of the reconstructed c class, $S_c = [\hat{z}_k^c] \in \mathbb{R}^{kr\times d}$, where $k \in [1, \cdots, K]andc \in [1, \cdots, C]$, and reconstructed query feature $Q_i = \hat{z}_i \in \mathbb{R}^{r\times d}$, where $i \in [1, \cdots, C \times M]$. S_c multiplies by weights W_γ^Q, W_γ^K and W_γ^V, repectively, obtaining S_C^Q, S_C^K, S_C^V, where $W_\gamma^Q, W_\gamma^K, W_\gamma^V \in \mathbb{R}^{d\times d}$. Similarly, Q_i multiplies by weights W_γ^Q, W_γ^K and W_γ^V, repectively, obtaining Q_i^Q, Q_i^K, Q_i^V.

The i query feature $\hat{Q}$ reconstructed from the c class support feature S_c^V and the c class support feature $\hat{S}$) reconstructed from the i query were calculated by (7) and (8),

$$\hat{Q} = \sigma\left(Q_i^Q (S_c^K)^T\right) S_c^V, \hat{Q} \in \mathbb{R}^{r\times d} \tag{7}$$

$$\hat{S} = \sigma\left(S_c^Q \left(Q_i^K\right)^T\right) Q_i^V, \hat{S} \in \mathbb{R}^{kr\times d} \tag{8}$$

The last part is the Euclidean distance metric (EDM). The distance between the query sample Q_i and the support sample in class c and the distance between the support sample in class c and the query sample Q_i are called d_{QS}, and d_{SQ} . And d_{QS}, and d_{SQ} were calculated using (9) and (10),

$$d_{QS} = ||Q_i^V - \hat{Q}||^2 \tag{9}$$

$$d_{SQ} = ||S_c^V - \hat{S}||^2 \tag{10}$$

where $||\cdot||$ is the Euclidean distance, λ_1 and λ_2 are learnable weight parameters, and both of them are initialized as 0.5. τ is a learnable temperature factor. The total distance d_i^c was calculated by (11). Then normalize to get $\hat{d}_i^c$, which was calculated using (12),

$$d_i^c = \tau(\lambda_1 d_{QS} + \lambda_2 d_{SQ}) \tag{11}$$

$$\hat{d}_i^c = \frac{\mathrm{e}^{-d_i^c}}{\sum_{c=1}^{C} \mathrm{e}^{-d_c^i}} \tag{12}$$

Based on $\hat{d}_i^c$, the total loss in one C-way K-shot task was calculated using (13),

$$L = -\frac{1}{M \times C} \sum_{i=1}^{M\times C} \sum_{c=1}^{C} \mathbf{1}(y_i = c) \log \hat{d}_i^c \tag{13}$$

III. Experimental Results and Analysis

A. Datasets

For performance evaluation, we assessed our method on three benchmark fine-grained datasets: CUB-200-2011 (CUB) [28], consisting of 11,788 images across 200 bird species; Stanford-Dogs (Dogs) [29], comprising 20,580 annotated images representing 120 dog breeds worldwide; and Stanford-Cars (Cars) [30], a dataset featuring 16,185 images covering 196 car types. All images were resized to 84×84.

TABLE I

Results on 1-shot and 5-shot tasks in the Conv4 backbone. The datasets we use are:CUB-200-2011 (CUB), Stanford-Dogs (Dogs) and Stanford-Cars (Cars).

Model	CUB		Dogs		Cars	
	1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet (NerulPS 17') [20]	64.82±0.23	85.74±0.14	46.66±0.21	70.77±0.16	50.88±0.23	74.89±0.18
Relation (CVPR 18') [21]	63.94±0.92	77.87±0.64	47.35±0.88	66.20±0.74	46.04±0.91	68.52±0.78
DN4 (CVPR 19') [22]	57.45±0.89	84.41±0.58	39.08±0.76	69.81±0.69	34.12±0.68	87.47±0.47
PARN (ICCV 19') [23]	74.43±0.95	83.11±0.67	55.86±0.97	68.06±0.72	66.01±0.94	73.74±0.70
SAML (ICCV 19') [24]	65.35±0.65	78.47±0.41	45.46±0.36	59.65±0.51	61.07±0.47	88.73±0.49
DeepEMD (CVPR 20') [25]	64.08±0.50	80.55±0.71	46.73±0.49	65.74±0.63	61.63±0.27	72.95±0.38
LRPABN (TMM 21') [1]	63.63±0.77	76.06±0.58	45.72±0.75	60.94±0.66	60.28±0.76	73.29±0.58
BSNet (TIP 21') [26]	62.84±0.95	85.39±0.56	43.42±0.86	71.90±0.68	40.89±0.77	86.88±0.50
CTX (NeurrIPS 20') [9]	72.61±0.21	86.23±0.14	57.86±0.21	73.59±0.16	66.35±0.21	82.25±0.14
FRN (CVPR 21') [19]	74.90±0.21	89.39±0.12	60.41±0.21	79.26±0.15	67.48±0.22	87.97±0.11
BiFRN (AAAI 23') [27]	79.08±0.20	92.22±0.10	64.74±0.22	81.29±0.14	75.74±0.20	91.58±0.09
Ours	**79.76±0.20**	**92.57±0.10**	**65.30±0.22**	**81.30±0.15**	**77.22±0.19**	**91.85±0.09**

TABLE II

Results on 1-shot and 5-shot tasks in the ResNet12 backbone. The datasets we use are:CUB-200-2011 (CUB), Stanford-Dogs (Dogs) and Stanford-Cars (Cars).

Model	CUB		Dogs		Cars	
	1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet (NerulPS 17') [20]	81.02±0.20	91.93±0.11	73.81±0.21	87.39±0.12	85.46±0.19	95.08±0.08
CTX (NeurIPS 20') [9]	80.39±0.20	91.01±0.11	73.22±0.22	85.90±0.13	85.03±0.19	92.63±0.11
DeepEMD (CVPR 20') [25]	75.59±0.30	88.23±0.18	70.38±0.30	85.24±0.18	80.62±0.26	92.63±0.13
FRN (CVPR 21') [19]	84.30±0.18	93.34±0.10	76.76±0.21	**88.74±0.12**	88.01±0.17	95.75±0.07
BiFRN (AAAI 23') [27]	85.44±0.18	94.73±0.09	76.89±0.21	88.27±0.12	**90.44±0.15**	**97.49±0.05**
Ours	**86.16±0.17**	**94.82±0.07**	**76.89±0.21**	88.72±0.12	89.77±0.15	97.22±0.06

B. Implementation Details

In our experiments, we tested two widely used backbone architectures: Conv-4 [19] and ResNet-12 [19]. The experiments were conducted using PyTorch on a single NVIDIA GeForce RTX 4090 GPU. We trained Conv-4 and ResNet-12 models for 1200 epochs with a weight shrinkage of 0.9, an initial learning rate of 0.1, and a 10x learning rate reduction every 400 epochs. For Conv-4 models, we employed 30-way 5-shot training, while for ResNet-12 models, we used 15-way 5-shot training. Regular data augmentation techniques such as center cropping, random horizontal flipping, and color jittering were applied to enhance stability.

C. Experimental Results

We use Conv-4 and ResNet-12 as the backbone of all compared models and test 5-way 1-shot and 5-way 1-shot classification performance. The outcomes utilizing Conv-4 and ResNet-12 as backbone networks are detailed in Table I and Table II, correspondingly.

In Table I, we can see that when Conv-4 is adopted, our method achieves the highest accuracy on all three datasets. In Table II, Apart from the result on the cars [30] dataset where performance falls slightly behind the BiFRN method when the ResNet-12 is adopted, our method achieves the highest accuracy.

In summary, compared with other newly proposed methods, our method has achieved stable and excellent performance on three fine-grained image datasets for 5-way 1-shot and 5-way 1-shot classification tasks. This is mainly due to our network. AAFRM enhances feature expressiveness, facilitates global contextual information capture within feature maps, and enables weighted feature fusion across various positions.

D. Ablation Study

To evaluate our method and model components, we employed Conv-4 and ResNet-12 as backbones and conducted ablation studies on three datasets.

First, we verify the validity of our approach by removing some model components. Then, we run the two model components, FSRM and AAFRM, in parallel to verify the effectiveness of the method. As shown in Table III, we remove FSRM and AAFRM, respectively, and run FSRM and AAFRM in parallel. The experimental results in Table III show that our fusion approach's model performance is further improved when using Conv-4 as the backbone network. However, when using ResNet-12 as the backbone network, the model performance is only improved on a portion of the dataset.

TABLE III
RESULTS ON 1-SHOT AND 5-SHOT TASKS WITH DIFFERENT BACKBONES ON CUB, DOGS, AND CARS DATASETS.

Backbone	Method	CUB		Dogs		Cars	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Conv-4	Baseline (ProtoNet)	64.82±0.23	85.74±0.14	46.66±0.21	70.77±0.16	50.88±0.23	74.89±0.18
	(FSRM)	79.08±0.20	92.22±0.10	64.74±0.22	81.29±0.14	75.74±0.20	91.58±0.09
	(AAFRM)	78.21±0.20	91.82±0.11	64.61±0.22	81.25±0.14	75.40±0.20	91.04±0.10
	(AAFRM, FSRM)	78.74±0.20	91.83±0.11	63.98±0.22	81.17±0.14	74.56±0.20	90.70±0.10
	(Ours)	**79.76±0.20**	**92.57±0.10**	**65.30±0.22**	**81.30±0.15**	**77.22±0.19**	**91.85±0.09**
ResNet-12	Baseline (ProtoNet)	81.02±0.20	91.93±0.11	73.81±0.21	87.39±0.12	85.46±0.19	95.08±0.08
	(FSRM)	85.44±0.18	94.73±0.09	76.89±0.21	88.27±0.12	**90.44±0.15**	**97.49±0.05**
	(AAFRM)	85.14±0.18	94.78±0.08	76.89±0.21	88.76±0.12	90.14±0.15	97.34±0.05
	(AAFRM, FSRM)	85.18±0.18	94.48±0.09	76.80±0.21	**88.93±0.11**	89.75±0.15	97.45±0.05
	(Ours)	**86.16±0.17**	**94.82±0.07**	**76.89±0.21**	88.72±0.12	89.77±0.15	97.22±0.06

IV. CONCLUSIONS

In this paper, we integrated an agent attention Model to enhance feature expressiveness, facilitate global contextual information capture within feature maps, and enable weighted feature fusion across various positions. Extensive experiments show that the proposed method performs well in three fine-grained image datasets and is very competitive.

REFERENCES

[1] 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 Transactions on Multimedia*, vol. 23, pp. 1666–1680, 2020.

[2] X.-S. Wei, Y.-Z. Song, O. Mac Aodha, *et al.*, "Fine-grained image analysis with deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 8927–8948, 2021.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[7] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 357–366.

[8] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8808–8817.

[9] C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: Spatially-aware few-shot transfer," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 981–21 993, 2020.

[10] W. Wang, E. Xie, X. Li, *et al.*, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[11] A. Hassani, S. Walton, J. Li, S. Li, and H. Shi, "Neighborhood attention transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6185–6194.

[12] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4794–4803.

[13] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. Lau, "Biformer: Vision transformer with bi-level routing attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 10 323–10 333.

[14] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning*, PMLR, 2020, pp. 5156–5165.

[15] Y. Xiong, Z. Zeng, R. Chakraborty, *et al.*, "Nyströmformer: A nyström-based algorithm for approximating self-attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 14 138–14 148.

[16] X. Sun, H. Xv, J. Dong, H. Zhou, C. Chen, and Q. Li, "Few-shot learning for domain-specific fine-grained image classification," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3588–3598, 2020.

[17] Y. Zhu, C. Liu, S. Jiang, *et al.*, "Multi-attention meta learning for few-shot fine-grained image recognition.," in *IJCAI*, Beijing, 2020, pp. 1090–1096.

[18] C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: Spatially-aware few-shot transfer," *Advances*

in Neural Information Processing Systems, vol. 33, pp. 21 981–21 993, 2020.

[19] D. Wertheimer, L. Tang, and B. Hariharan, "Few-shot classification with feature map reconstruction networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8012–8021.

[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[21] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[22] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. 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*, 2019, pp. 7260–7268.

[23] Z. Wu, Y. Li, L. Guo, and K. Jia, "Parn: Position-aware relation networks for few-shot learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6659–6667.

[24] F. Hao, F. He, J. Cheng, L. Wang, J. Cao, and D. Tao, "Collect and select: Semantic alignment metric learning for few-shot learning," in *Proceedings of the IEEE/CVF international Conference on Computer Vision*, 2019, pp. 8460–8469.

[25] C. Zhang, Y. Cai, G. Lin, and C. 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*, 2020, pp. 12 203–12 213.

[26] X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, and J.-H. Xue, "Bsnet: Bi-similarity network for few-shot fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 1318–1331, 2020.

[27] J. Wu, D. Chang, A. Sain, *et al.*, "Bi-directional feature reconstruction network for fine-grained few-shot image classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 2821–2829.

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

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

[30] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.