2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506685

COUPLED PATCH SIMILARITY NETWORK FOR ONE-SHOT FINE-GRAINED IMAGE RECOGNITION

Sheng Tian *Hao Tang* *Longquan Dai**

School of Computer Science and Engineering, Nanjing University of Science and Technology

ABSTRACT

One-shot fine-grained image recognition (OSFG) aims to distinguish different fine-grained categories with only one training sample per category. Previous works mainly focus on learning a global feature representation through only a using single similarity metric branch, which is unsuitable for OSFG to effectively capture subtle and local differences under limited supervision. In this work, we propose a Coupled Patch Similarity Network (CPSN) for OSFG. Firstly, we propose a Feature Enhancement Module (FEM) to extract more discriminative features of the fine-grained samples. Then, we develop two coupled and symmetrical branches to capture discriminative parts of the samples and reduce the deviation of the distance metric. For each branch, we design a Patch Similarity Module (PSM) to calculate the patch similarity map for the sample pair. Especially, a Patch Weight Generator (PWG) is proposed to generate the patch weight map, which indicates the degree of importance for each position in the patch similarity map, so that the model can focus on diverse and informative parts. We analyze the effect of the different components in the proposed network, and extensive experimental results demonstrate the effectiveness and superiority of the proposed method on two fine-grained benchmark datasets.

Index Terms— one-shot, fine-grained, image recognition

1. INTRODUCTION

Fine-grained recognition [1, 2, 3] aims to identify the subcategories belonging to the same entry-level category while having slight and subtle differences, which has attracted extensive attention recently. In this work, we study traditional fine-grained image recognition in a more challenging setting, called one-shot fine-grained recognition (OSFG), which aims to correctly identify novel fine-grained categories in one-shot learning scenarios (i.e., only one training example per category). Especially, OSFG not only requires capturing the subtle differences among categories but also shows the excellent generalization ability beyond the limited supervision [4]. Obviously, the generalized few-shot learning algorithms [5, 6, 7] are also unsuitable for this task.

Many recent approaches have been proposed to address this challenging task. Wei et al. [8] design a piecewise mapping classifier by employing the bilinear features. LRPABN [9] proposes a pairwise bilinear pooling operator to perform feature alignment for learning an efficient distance metric. To capture more discriminative features, Zhu *et al.* [10] develop a multi-attention meta-learning method to learn informative parts by attention mechanism. Similarly, BSNet [11] is proposed to improve the generalization ability by decreasing the similarity bias and extracting more diverse features. Almost all of the above approaches have a commonality that global feature representations are directly used to compute the similarity scores of the query samples relative to the support samples for performing instance-level classification. However, fine-grained categories are distinguished by subtle parts, where local feature representations may be more effective to capture diverse and informative parts for OSFG.

In this work, we propose a *Coupled Patch Similarity Network (CPSN)* for OSFG to capture discriminative parts of sample pairs and reduce the deviation of the distance metric. As shown in Fig. 1 (a), we first propose a *Feature Enhancement Module (FEM)* to enhance the features of sample pair by cross-channel information, which is inserted into the top layer of the feature extractor. Then, the fine-grained relations in the pair of enhanced features can be captured by two coupled branches. For each branch, we leverage a *Patch Similarity Module (PSM)* to compute the patch similarity map between the corresponding patch features in the sample pair. Specifically, we propose a *Patch Weight Generator (PWG)* to generate a weight map, which indicates the degree of importance for each position in the patch similarity map, so that the model can focus on diverse and informative parts between sample pair. Finally, the final similarity score for the input sample pair is obtained by combining the patch similarity maps with the corresponding weight maps of the two branches. Notice that we also add a global branch to perform patch-level classification overall categories during the training stage, which utilizes spatial information of the last feature map as a regularization to constraint the feature extractor. We validate the performance of the proposed method for one-shot fine-grained image recognition on two benchmark datasets, and experimental results show that our method achieves state-of-

*Corresponding author.

This work was supported by the National Natural Science Foundation of China under Grant No. 62072238.

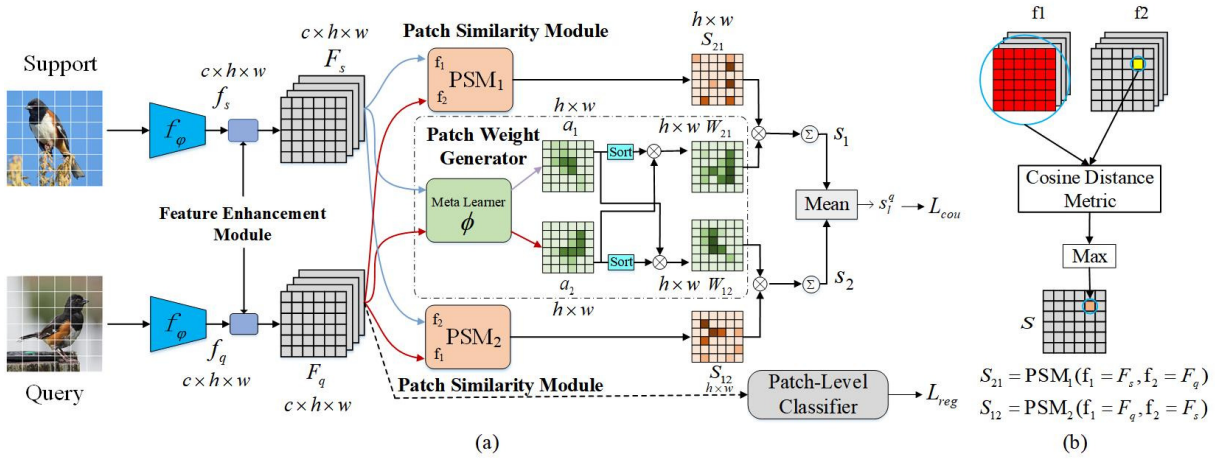

Fig. 1. (a) The network architecture of the proposed CPSN. The final similarity score between the support sample and query sample is only obtained by integrating two weighted patch similarity maps of two branches. (b) Patch Similarity Module.

the-art results and largely outperforms all baselines.

2. PROPOSED METHOD

In this section, we first define the problem of one-shot fine-grained image recognition. Then we illustrate the network architecture and the loss functions of CPSN. Finally, we describe our modules and the roles they play in detail.

2.1. Problem Definition

During the training phase, we first randomly sample N categories from the entire set of base categories C_{base} and then sample 1 and M samples from each category for the support set $S = (x_s, y_s)_{s=1}^{N\times 1}$ and query set $Q = (x_q, y_q)_{q=1}^{N\times M}$, where $y_s, y_q \in \{1, \ldots, N\}$ are the category label. Similarly, the test phase of OSFG is also to perform C-way 1-shot classification but sample C categories from the novel categories C_{novel}. Notice that C_{base} is disjoint with C_{novel}.

2.2. Network Architecture

The network architecture of the proposed method is presented in Fig. 1 (a). We can observe that there is a global branch to perform patch-level classification, in addition to two coupled branches to measure the patch similarity between a sample pair including a support sample x_s and a query sample x_q. Note, the first branch aims to get the similarity score s_1 of x_q relative to x_s, where the opposite is that the second branch aims to get the similarity score s_2 of x_s relative to x_q. Here, due to the symmetry of two coupled branches, we just introduce the process for forward-propagating of the first branch. Firstly, f_s, f_q are denoted as the feature representations of x_s, x_q generated by the feature extractor f_φ. Then, they are enhanced by our FEM, and the enhanced features are written as F_s, F_q. Next, we can get a patch similarity map S_{21} and a patch weight map W_{21} generated by our PSM and PWG, respectively. Finally, the similarity score s_1 of the first branch can be calculated by firstly weighting S_{21} by W_{21} and then summing scores of all patches in S_{21}. So similarly, we can get the similarity score s_2 of x_s relative to x_q from the second branch. So the final similarity score s_l^q of input sample pair is obtained by meaning the similarity scores of two branches.

2.3. Loss Function

The top loss L_{cou} in Fig. 1 is a cross-entropy loss function that aims to classify query sample x_q with the similarity score s_l^q, where q means the q^{th} query sample, l means the l^{th} category in an OSFG task, so L_{cou} can be formulated as follow:

$$L_{cou} = -\sum_{q=1}^{N\times M} \log \frac{\exp(s_l^q)}{\sum_{k'} \exp(s_{k'}^q)}, k' \in \{0, 1, ..., N\}, \quad (1)$$

The bottom loss L_{reg} is acted as a regularization by using a patch-level classifier to force the model to classify all patches over all training categories correctly in the query feature map $F_q \in \mathbb{R}^{c\times h\times w}$. So L_{reg} can be formulated as follow:

$$L_{reg} = -\sum_{q=1}^{N\times M}\sum_{j=1}^{h\times w} \log\Big(\big(\mathrm{Softmax}\big(W_c(F_q)_j\big)\big)_{t^q}\Big), \quad (2)$$

where $t^q \in \{1, 2, ..., t\}$ is the true semantic category label of x_q, $W_c \in \mathbb{R}^{t\times c}$ is the weight of the classifier. We get the final loss L_{total} by utilizing an adjustable coefficient β to integrate the L_{cou}, L_{reg}, as shown below:

$$L_{total} = L_{cou} + \beta * L_{reg}, \quad (3)$$

2.4. Feature Enhancement Module

We design a feature enhancement module (FEM) to obtain more discriminative feature by recalibrating channel-wise feature response. For example, given a support feature map $f_s \in \mathbb{R}^{c\times h\times w}$, we first obtain a matrix $M_s \in \mathbb{R}^{c\times 1\times 1}$ by summing it along the spatial dimension. Then, we apply a softmax function to the flattened matrix M_s and the normalized matrix is used to weight the feature map f_s. Therefore, the final output F_s of FEM can be formulated as follow:

$$F_s = \Big(1 + \mathrm{Softmax}(M_s)\Big) f_s, \quad (4)$$

Significantly, our proposed FEM is a **parameter-free** layer which is inserted after the last layer of the feature extractor to avoid excessive cost.

2.5. Patch Similarity Module

As shown in Fig. 1 (a), a patch similarity module (PSM) is proposed to obtain a more precise similarity score between x_s and x_q by taking more local information into account. specifically, there are two symmetrical PSM in our CPSN to reduce the deviation of the distance metric, where PSM_1 aims to compute the patch similarity map of F_q relative to F_s and PSM_2 aims to compute the patch similarity map of F_s relative to F_q. Here, taken PSM_1 as example and the process is shown in Fig. 1 (b). We first reshape two enhanced feature maps $F_s, F_q \in \mathbb{R}^{c\times h\times w}$ with $m = h \times w$ patches (i.e., $F_s = [p_1, p_2, \ldots, p_{h\times w}], F_q = [q_1, q_2, \ldots, q_{h\times w}]$). Then, a cosine distance metric and a maximization operation are applied to F_s and F_q in order and generate a patch similarity map $S_{21} \in \mathbb{R}^{h\times w}$, where each element of S_{21} indicates the maximum similarity value between a patch in F_q and all patches in F_s. Similarly, we can also obtain another patch similarity map S_{12} by PSM_2. We denote q_j as the j^{th} patch in F_q and p_i as the i^{th} patch in F_s , each element of S_{21} and S_{12} can be formulated as follow:

$$\begin{aligned} \mathrm{S}_{21}^{j} &= \max[\cos(q_j, p_i)_{i=1,\ldots,h\times w}], \\ \mathrm{S}_{12}^{i} &= \max[\cos(p_i, q_j)_{j=1,\ldots,h\times w}], \end{aligned} \tag{5}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity. Meanwhile, we denote K_{21}, K_{12} as the maximum index maps, which indicate the corresponding positions that can obtain the maximum similarity in F_s and F_q, respectively. Furthermore, any one of the elements in S_{21} might show how similar their foreground patches or their background patches are, when given two fine-grained samples.

2.6. Patch Weight Generator

Additionally, a patch weight generator (PWG) is designed to equip the PSM with the ability by which it can pay more attention to the discriminative foreground patches rather than the clutter background patches in the patch similarity map S_{21} or S_{12}. In Fig. 1 (a), we first make the enhanced feature maps $F_s, F_q \in \mathbb{R}^{c\times h\times w}$ go through by a meta learner ϕ that consists of two convolution layers to get the attention maps $a_1 = \phi(F_s), a_2 = \phi(F_q) \in \mathbb{R}^{h\times w}$, which assign each patch a value indicating the degree of importance for that location. Namely, the value corresponding to the background patch in a_1 is lower, while the foreground patch is higher. Then two patch weight maps W_{21}, W_{12} are generated to match S_{21}, S_{12} according to the corresponding index maps K_{21}, K_{12}:

$$\mathrm{W}_{21} = a_2 \odot [a_1, K_{21}], \quad W_{12} = a_1 \odot [a_2, K_{12}], \tag{6}$$

where $\odot$ denotes element-wise multiplication and $[a, K]$ indicates that a sort operation is applied to a by an index map K. Understandably, if the values of the spatial positions of background patches in a_2 and sorted $[a_1, K_{21}]$ are small, the final results in the corresponding positions of W_{21} will be smaller by element-wise multiplication. We can obtain two similarity scores s_1, s_2 by first applying element-wise multiplication between the patch similarity map and the corresponding patch weight map and then a summation over all patches, as following:

$$\mathrm{s}_1 = \sum_{j=1}^{h\times w} S_{21}^{j} \times W_{21}^{j}, \quad \mathrm{s}_2 = \sum_{\mathrm{i}=1}^{\mathrm{h\times w}} \mathrm{S}_{12}^{\mathrm{i}} \times \mathrm{W}_{12}^{\mathrm{i}}. \tag{7}$$

3. EXPERIMENTS

3.1. Datasets

We conduct experiments on two widely used fine-grained datasets: CUB-200-2011 [12] and Stanford Cars [13]. CUB-200-2011 is a fine-grained bird dataset that contains a total of 11788 bird images from 200 species. We follow the evaluation setup in [14, 15] to split 100, 50, and 50 categories as training set, validation set, and testing set, respectively. Stanford Cars is a fine-grained car dataset that contains a total of 16185 car images from 196 categories. We follow the separation in [16, 10] to split 130 categories for training, 17 categories for validation, and 49 categories for testing.

3.2. Implementation Details

We conduct experiments with the proposed CPSN on 5-way 1-shot settings. Following the experimental settings described in Sec. 2.1, we perform 2000 random episodes in the testing phase, each of which has 15 query samples per category, and reports mean average accuracy with the 95% confidence interval. For a fair comparison with state-of-the-art methods, two widely used backbones are adopted as feature extractor in experiments, which contain a shallow network Conv-64 as in [14, 15] and a deeper network ResNet-12 as in [17]. The input size of both networks is 84×84. Our method utilizes the SGD optimizer with an initial learning rate of 0.1 and weight decay of 0.0005 to train from scratch. Notice that the coefficient β in the Eq. 3 is set to 0.1 for all experiments.

3.3. Comparison with State-of-the-Arts

We use the CUB-200-2011 dataset to conduct main experiments. Table 1 presents the mean accuracy of different methods with different backbones (i.e., Conv-64 and ResNet12). An observation is that the proposed CPSN outperforms the previous approaches at a larger margin. As shown in Table 1, for shallow network Conv-64, our CPSN gains **15.18%**, **19.25**%, **10.72**%, **12.01**%, **6.08**%, **3.09**% improvements over three classical methods [5] [19] [18] and three relatively new methods [16] [9] [10], respectively. Similarly, for deeper network ResNet12, the proposed CPSN obtains significant improvements compared with start-of-the-arts including [5] [19] [18] [14] [20]. Specially[1], our

[1]ResNet18 has deeper network and higher resolution (i.e., 224×224)

Table 1. Average accuracy (%) comparison to state-of-the-arts with 95% confidence intervals on CUB-200-2011. * denotes that it is implemented with our setting.

Methods	Ref.	Backbone	5-way 1-shot
MatchNet [18]	NeurIPS'16	Conv-64	60.52±0.88
ProtoNet [5]	NeurIPS'17	Conv-64	50.46±0.88
RelationNet [19]	CVPR'18	Conv-64	58.99±0.52
DN4* [16]	CVPR'19	Conv-64	57.70±0.98
LRPABN [9]	TMM'20	Conv-64	63.63±0.77
MattML [10]	IJCAI'20	Conv-64	66.29±0.56
CPSN	Ours	Conv-64	**69.38±0.54**
MatchNet [18]	NeurIPS'16	ResNet-12	60.96±0.35
ProtoNet [5]	NeurIPS'17	ResNet-12	71.57±0.89
RelationNet [19]	CVPR'18	ResNet-12	60.21±0.35
Baseline++ [14]	ICLR'19	ResNet-12	67.30 ±0.86
MetaOptNet [20]	CVPR'19	ResNet-12	75.15 ±0.46
Neg-Cosine [21]	ECCV'20	ResNet-18	72.66 ±0.85
Centroid-A [22]	ECCV'20	ResNet-18	74.22±1.09
CPSN	Ours	ResNet-12	**77.93±0.49**

Table 2. Average accuracy (%) comparison to state-of-the-arts with 95% confidence intervals on Stanford Cars.

Methods	Ref.	Backbone	5-way 1-shot
MatchNet [18]	NeurIPS'16	Conv-64	44.73±0.77
ProtoNet [5]	NeurIPS'17	Conv-64	36.54±0.74
RelationNet [19]	CVPR'18	Conv-64	47.79±0.49
DN4 [16]	CVPR'19	Conv-64	61.51±0.85
LRPABN [9]	TMM'20	Conv-64	60.28±0.76
MattML [10]	IJCAI'20	Conv-64	66.11±0.54
CPSN	Ours	Conv-64	**71.17±0.49**
CPSN	Ours	ResNet-12	**84.27±0.44**

CPSN also defeats [22] and [21] that use ResNet-18 as feature extractor by **5**% and **7.25**% gains, which demonstrates the superiority of the coupled patch similarity network. We use another Stanford Cars dataset for secondary experiments. As shown in Table 2 , our CPSN achieves **5.06**% gains over the start-of-the-art MattML[10] and shows absolute advantages compared with other methods. When using the deeper network ResNet12, our CPSN also achieves a pretty performance **84.23**%, but there are no comparable methods for this dataset, unfortunately.

3.4. Further Analysis

Ablation Studies. Table 3 shows ablation studies of the 5-way 1-shot settings on CUB-200-2011 dataset when using ResNet12. Here, PSM_1, PSM_2 and REG denote the two branches of our PSM and global regularized branch, respectively. The baseline method only has a single branch PSM_1 and does not employ other modules. Only combining PSM_1 with PWG, we gain 33.08% improvements, which illustrates that PWG is very useful for PSM. On this basis, the performance can be improved steadily by adding coupled branch PSM_2, global branch REG, and FEM, which demonstrates the effectiveness of our contributions.

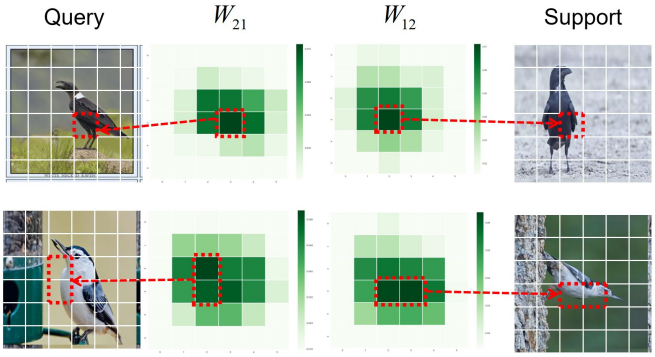

Fig. 2. Visualization results. The green color in the patch weight maps W_{21} and W_{12} indicate weights for the corresponding patches in support and query examples. The darker of the patch indicates that the model pays more attention to this patch when making decisions.

Table 3. Ablation test results of CPSN on CUB-200-2011.

PSM_1	PWG	PSM_2	REG	FEM	5-way 1-shot
✓					42.31±0.53
✓	✓				75.39±0.52
✓	✓	✓			76.24±0.50
✓	✓	✓	✓		76.81±0.50
✓	✓	✓	✓	✓	**77.93±0.49**

Visualization. Fig. 2 shows the visualization results about the corresponding patch weight maps generated by the PWG for the support sample and query sample, respectively. In these visualizations, highlighted regions in weight maps are relevant to the discriminative patch in the samples. More specifically, the salient part in the first row is the bird's wings, and the salient part in the second row is the bird's abdomens. This demonstrates that, due to the help of weight maps generated from PWG, our model can be adaptive to focus on discriminative patches and makes the final classification decision by comparing these important patches.

4. CONCLUSION

In this work, we propose a Coupled Patch Similarity Network (CPSN) for one-shot fine-grained image recognition (OSFG), which can capture discriminative parts of sample pairs and reduce the deviation of the distance metric. Firstly, we propose a feature enhancement module to extract more discriminative features of the samples. Then, we develop two coupled branches, each contains a patch similarity module and a patch weight generator, which are designed to compute the similarity scores of the sample pairs better. Finally, we integrate the similarity scores of the two branches to make the final decision for OSFG. Extensive experimental results on two major fine-grained benchmark datasets demonstrate that the proposed method achieves state-of-the-art performance compared with previous methods.

5. REFERENCES

[1] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian, "Picking deep filter responses for fine-grained image recognition," in *CVPR*, 2016, pp. 1134–1142.

[2] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *ICCV*, 2017, pp. 5209–5217.

[3] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim, "Cross-x learning for fine-grained visual categorization," in *ICCV*, 2019, pp. 8242–8251.

[4] Zhang Dong, Zhang Hanwang, Tang Jinhui, Hua Xiansheng, and Sun Qianru, "Causal intervention for weakly supervised semantic segmentation," in *NeurIPS*, 2020.

[5] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017, pp. 4077–4087.

[6] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang, "Few-shot image recognition with knowledge transfer," in *ICCV*, 2019, pp. 441–449.

[7] Hao Tang, Zechao Li, Zhimao Peng, and Jinhui Tang, "Blockmix: meta regularization and self-calibrated inference for metric-based meta-learning," in *ACM MM*, 2020, pp. 610–618.

[8] Xiu-Shen Wei, Peng Wang, Lingqiao Liu, Chunhua Shen, and Jianxin Wu, "Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples," *TIP*, pp. 6116–6125, 2019.

[9] Huaxi Huang, Junjie Zhang, Jian Zhang, Jingsong Xu, and Qiang Wu, "Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification," *TMM*, 2020.

[10] Yaohui Zhu, Chenlong Liu, and Shuqiang Jiang, "Multi-attention meta learning for few-shot fine-grained image recognition," in *IJCAI*, 2020, pp. 1090–1096.

[11] Xiaoxu Li, Jijie Wu, Zhuo Sun, Zhanyu Ma, Jie Cao, and Jing-Hao Xue, "Bsnet: Bi-similarity network for few-shot fine-grained image classification," *TIP*, 2020.

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

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei, "3d object representations for fine-grained categorization," in *ICCVW*, 2013, pp. 554–561.

[14] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang, "A closer look at few-shot classification," in *ICLR*, 2019.

[15] Luming Tang, Davis Wertheimer, and Bharath Hariharan, "Revisiting pose-normalization for fine-grained few-shot recognition," in *CVPR*, 2020, pp. 14352–14361.

[16] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *CVPR*, 2019, pp. 7260–7268.

[17] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen, "Cross attention network for few-shot classification," in *NeurIPS*, 2019, pp. 4003–4014.

[18] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al., "Matching networks for one shot learning," in *NeurIPS*, 2016, pp. 3630–3638.

[19] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018, pp. 1199–1208.

[20] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto, "Meta-learning with differentiable convex optimization," in *CVPR*, 2019, pp. 10657–10665.

[21] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu, "Negative margin matters: Understanding margin in few-shot classification," *ECCV*, 2020.

[22] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné, "Associative alignment for few-shot image classification," in *ECCV*, 2020, pp. 18–35.