REPRESENTATIVE LOCAL FEATURE MINING FOR FEW-SHOT LEARNING

Kun Yan[1], *Lingbo Liu*[2], *Jun Hou*[3], *Ping Wang*[1,4,5,*]

[1]School of Software and Microelectronics, Peking University
[2]School of Data and Computer Science, Sun Yat-Sen University
[3]SenseTime Research
[4]National Engineering Research Center for Software Engineering, Peking University
[5]Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education

ABSTRACT

Few-shot learning aims to recognize unseen images of new classes with only a few training examples. While great progress has been made with deep learning technology, most metric-based works rely on the measurement based on global feature representation of images, which is sensitive to background factors due to the scarcity of training data. Given this, we propose a novel method that chooses representative local features to facilitate few-shot learning. Specifically, we propose a "task-specific guided" strategy to mine local features that are task-specific and discriminative. For each task, we first mine representative local features for labeled images by a loss guided mechanism. Then these local features are used to guide a classifier to mine representative local features for unlabeled images. In this way, task-specific representative local features can be selected for better classification. We empirically show our method can effectively alleviate the negative effect introduced by background factors. Extensive experiments on two few-shot benchmarks show the effectiveness of the proposed method.

Index Terms— Few-shot learning, representative local features, feature mining, metric-based learning

1. INTRODUCTION

In recent years, deep neural networks [1–6] have made significant progress in the image classification task, due to their great capacities for learning knowledge from abundant labeled training examples. As a contrast, the human visual system can learn to recognize new classes with only a few labeled examples. Such a few-shot learning challenge has attracted great attention in research. However, in few-shot learning situation, the number of samples of each class is limited; therefore, it is difficult for the visual model to obtain the sample distribution of each class.

To tackle the few-shot learning problem, various methods have been put forward. A recently effective solution [7–11] is to train a model on the seen classes to learn a generalizable feature embedding space, where can determine the similarity of two images according to their corresponding feature embeddings by utilizing a metric (like Euclidean distance), and further apply it to images in unseen classes. This kind of method falls into the metric-based paradigm. Although great progress has been made, metric-based methods generally rely on global feature representation of images, which ignore spatial information. As this type of representation is mixed with too many background factors, the model may be not robust to this unrelated information due to the scarcity of labeled examples in few-shot learning.

To address the aforementioned weakness, some recent works [12–15] resort to the local feature representation of images. Different from methods rely on the global feature, local feature based methods will take a further step to more fine-grained local information, which can highlight semantical parts. However, existing local feature based methods suffer from mainly two limitations: (1) Discovered knowledge about local features from seen classes are equally effective for any tasks which comprise of unseen classes. We argue that each task should explore discriminative knowledge specialized for this task in addition to the generalized one from seen classes for better classification. (2) They use the information of all local features contain no matter semantical parts or background factors. Although some works [14, 15] have put more weight on semantical parts and less weight on background parts through different attention mechanisms, background factors can still more or less confuse model in the situation of a few labeled samples.

Towards this, we propose a "task-specific guided" strategy to find local features that are task-specific and representative. To support this strategy, we develop a Prototype Selection Module (PSM) and a Task Adaption Module (TAM) for labeled and unlabeled images respectively. Concretely, for each particular task, PSM first finds representative local features for labeled images by a loss guided mechanism through a simple image classification task. We assume image labels are strong supervision information, as they can directly reflect objects in images. Therefore, If a local feature can contribute to image classification, it is representative. If removing a local feature does not affect the corresponding image classification, then we think it useless. For example, given a 'cat' image, if we remove all local features which mainly contains background factors, and only keep local features mainly depict the 'cat' object. We can still classify this image into the 'cat' class. Through this finding, we further propose a local feature importance algorithm based on Taylor expansion to achieve loss guided mechanism easily. After getting representative local features of labeled images through PSM, TAM will use these features to adapt a classifier, especially for the current task. Through the guidance of local features from labeled images, the classifier will be able to select representative (discriminative) local features for unlabeled images in each particular task.

Through our "task-specific guided" strategy, we can further tailor the visual knowledge extracted from the seen classes to the unseen ones according to a particular task. In this way, we can generate local features more discriminative and representative as they are customized according to each particular task. What's more, we discard all local features which mainly contain background information at the

[*] means the corresponding author

same time. Therefore, we achieve to explicitly mitigate the negative effect of background factors. Extensive experiments on two few-shot benchmarks show the effectiveness of our method.

The main contributions of this work are summarized as two folds: (1) We develop a novel metric-learning based model for few-shot learning, which can adapt to each particular task to find representative local features and discard irrelevant ones. (2) We propose a "task-specific guided" strategy, which can guide the model find representative local features for labeled images first through a loss guided mechanism, and then further guide the model to find representative local features for unlabeled images in each particular task.

2. PRELIMINARIES

In few-shot learning (FSL), we are given a base class set $\mathcal{C}_b$ and a novel class set $\mathcal{C}_n$. Specifically, each class in $\mathcal{C}_b$ has sufficient labeled images, while only a few labeled samples are obtained for each class in $\mathcal{C}_n$. Note that $\mathcal{C}_b$ and $\mathcal{C}_n$ are disjoint. In this setting, the goal of FSL is to obtain a good classifier for novel classes.

Following the previous work [8], we adopt the episode-based training scheme to facilitate few-shot learning. In each episode, each classification task is performed on a support set $\mathcal{S}$ and query set $\mathcal{Q}$. In particular, $\mathcal{S}$ follows a N-way K-shot setting. N is the number of classes and K is the number of labeled examples in each class, where K is a small integer, such as 1 or 5. So the support set and query set can be respectively defined as $\mathcal{S} = \{(x_i^s, y_i^s)_{i=1}^{n_s}\}$ and $\mathcal{Q} = \{(x_i^q, y_i^q)_{i=1}^{n_q}\}$, where $n_s = N \times K$ and n_q are the sample numbers of support/query set. In training episodes, we optimize our model with $\mathcal{S}/\mathcal{Q}$ sampled from $\mathcal{C}_b$. During the testing episodes, we measure the generalization performance of a model with $\mathcal{S}/\mathcal{Q}$ sampled from $\mathcal{C}_n$, where labels in $\mathcal{S}$ are known and those in $\mathcal{Q}$ are unknown. The predicted category of a query image is determined by taking the class with the highest similarity score.

3. METHOD

In this work, we propose a novel framework for few-shot learning, which mines task-specific representative local features for images. As shown in Fig. 1, our framework consists of four components, including a CNN-based feature extractor, a prototype selection module (PSM), a task adaption module (TAM), and a similarity computation module. In particular, our whole framework is optimized according to a "task-specific guided" strategy. Specifically, given a task, our PSM first mines representative local features for support set according to a classification loss guided mechanism, based on the observation that given the strong supervision of labels, the more local vectors are conducive to correct image classification, the more representative they are. Then, our TAM utilizes the features selected from PSM to train a binary classifier to discover similar local features for the query set. In this way, we can automatically guide the model to find representative local features that are more suitable for each particular task.

3.1. CNN-based Feature Extractor

Given a support/query image X, we first feed it into a convolutional neural network (CNN) to extract image-level feature. For convenience, the feature of X is represented as $f_\theta(X) \in \mathbb{R}^{c \times h \times w}$, where $f_\theta(\cdot)$ is a CNN-based feature extractor (e.g., ResNet [4]), θ is the set of its corresponding learnable parameters, and c, h, w are the channel number, height, width of feature respectively. In this way, we can get m ($m = h \times w$) c-dimensional local features for the given image as: $f_\theta(X) = [\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_m]$. It should be noted that local features contain spatial information. Thus, we can mitigate the interference of irrelevant background factors by just selecting the most representative local features to represent the target objects.

3.2. Prototype Selection Module

The proposed PSM is used to mine a fixed number (k_{psm}) of representative local features for each image in the support set. We view representative local features of all labeled images of each class as its corresponding prototype. The most important factor that distinguishes each task is the image category, as different tasks have different image categories. Therefore, if we can make the model more suitable for selecting representative local features of the current task categories, it is equivalent to make the model adapt to this task. Further, image label is a strong supervision information as it reflects image category directly, and image classification is the most relevant task. Based on these inspirations, PSM is modeled as a simple classification task. Intuitively, if a local feature is helpful for image classification, it is representative. To select local features efficiently, we further propose a **local feature importance algorithm** based on Taylor expansion

We can use the impact on the classification loss to reflect whether a local feature is helpful for image classification. Specifically, if the loss value changes a lot when removing a local feature, we argue this local feature is important for classification. In other words, this local feature is representative as it contains the information of the target object. If removing a local feature has no effect on the loss value, we can think this local feature is useless. Towards this, for an image-level feature, we multiply each local feature by a factor ρ. When the ρ is equal to 0, it is equivalent to remove the corresponding local feature, and if it is 1, it is equivalent to keep this local feature. To ensure that the model can be updated by gradient descent, we limit ρ to continuous values in the interval of 0 to 1 ($\rho \in [0, 1]$) as a trainable scaling factor. Then we can define the function to evaluate the importance of a local feature according to the impact on the classification loss as:

$$g(\rho) = |\Delta\mathcal{L}_\Omega| = |\mathcal{L}_\Omega(\rho) - \mathcal{L}_\Omega(0)|, \tag{1}$$

where $\mathcal{L}$ means the classification loss value, where Ω includes all examples and parameters in PSM except ρ.

We can further get a simpler and easier form to implement by applying the Taylor expansion, which is:

$$\mathcal{L}_\Omega(x) = \mathcal{L}_\Omega(\rho) + \frac{\mathcal{L}_\Omega^{(1)}(\rho)}{1!}(x - \rho) + ... + \frac{\mathcal{L}_\Omega^{(n)}(\rho_0)}{n!}(x - \rho)^n + R_n(x).$$

We then estimate $\mathcal{L}_\Omega(0)$ as a function of ρ:

$$\mathcal{L}_\Omega(0) = \mathcal{L}_\Omega(\rho) - \rho\mathcal{L}_\Omega^{(1)}(0) + R_1(\rho). \tag{2}$$

In this way, the Eq. (1) can be rewritten as (we remove the Lagrange remainder $R_1(\rho)$ for approximation) :

$$g(\rho) = |\rho\mathcal{L}_\Omega^{(1)}(\rho) - R_1(\rho)| \approx |\rho\mathcal{L}_\Omega^{(1)}(\rho)|. \tag{3}$$

Towards this end, we use $g(\rho)$ to represent the importance of each local feature. According to Eq. (3), $g(\rho)$ can be obtained by simply multiplying ρ and its gradient which can be calculated automatically in some machine learning frameworks, such as PyTorch.

In Fig. 1, we implement our PSM as a classifier with a one-layer fully-connected layer. The trainable scaling factor ρ for each local feature can be learned with m_{sub1} iterations of this classifier, where for m_{sub1}, a small number (such as 5) is enough due to the small number of training examples. To make the model more stable, we add up $g(\rho)$ in each iteration as the final importance score of a local

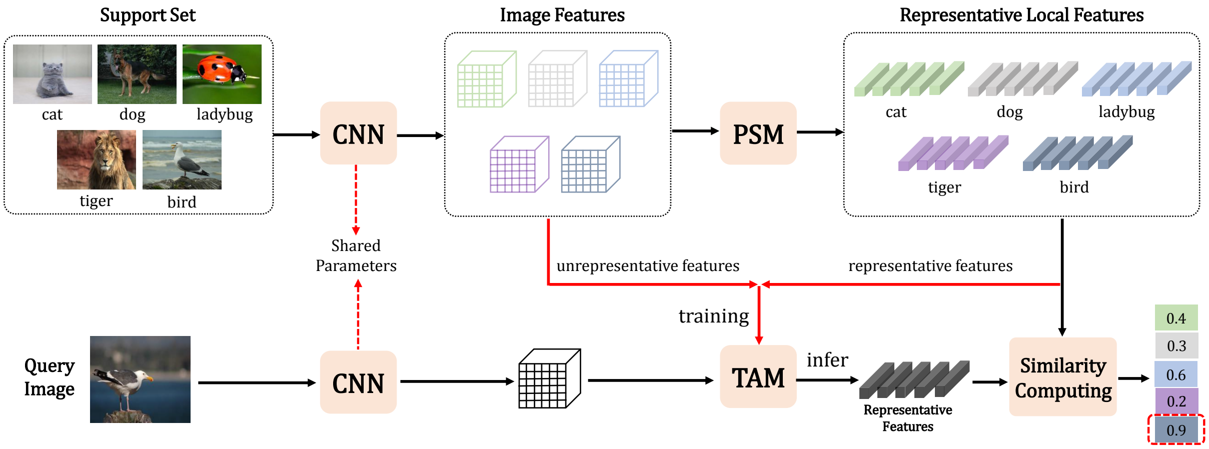

Fig. 1. Overview of the proposed framework for N-way K-shot few-shot learning task. Specifically, our framework consists of a CNN-based feature extractor, a prototype selection module (PSM), a task adaption module (TAM) and a similarity computation module.

feature. Then $g(\rho)$ can be described as:

$$g(\rho) = \sum_{i=1}^{m_{sub}} g_i(\rho_i) = \sum_{i=1}^{m_{sub}} |\rho_i \mathcal{L}_{\Omega}^{(1)}{}_i(\rho_i)|, \tag{4}$$

After getting the importance store for each local feature, we rank all local features in descending order according to their importance stores, followed by selecting the first k_{psm} local features for every image. Then in an N-way K-shot setting, we cluster $K * k_{psm}$ local features for each of N classes as its prototype.

3.3. Task Adaption Module

The proposed TAM is used to mine representative local features for query set images. In our work, TAM is implemented as a binary classifier with one fully-connected layer, and it learns to distinguish representative local features for query images. As mentioned in the Section 3.2, the key to adapt the model to a given task is the class category in this task. However, image labels in the query set are agnostic. What is certain is that the category of an image in the query set must belong to a certain category in the support set. As the PSM generates representative local features for support set specialized for the current task, we can view these features as the weak supervision information in a feature level, instead of the strong supervision information directly from labels.

Specifically, we define the subnet of TAM as a binary classifier. During classifier training, the input involves two kinds of local features. The first kind is from representative local features selected by PSM, these local features are viewed as positive samples; the second is from local features discarded by PSM, these local features mainly contain background information and are viewed as negative samples. The classifier will be trained to classify positive samples into label 1 and 0 for negative samples with m_{sub2} iterations. It is worth noting that m_{sub2} is also a small number just like m_{sub1} in PSM. Actually, in the experiment setting, both m_{sub1} and m_{sub2} are set to 5. Therefore, the subnet training process is efficient. During classifier inference, given an image-level feature, it would output a score between 0 to 1 for each local feature. Different from PSM that selects a fixed number of local features, our TAM would keep its all local features whose scores are greater than a defined threshold ϵ., because different objects have different sizes. Large objects may have more representative local features compared with relatively small objects that have less representative local features.

3.4. Similarity Computation

For convenience, the representative local features of a query example x^q are denoted as $[\mathrm{x}_1^q, \mathrm{x}_2^q, ..., \mathrm{x}_m^q] \in \mathbb{R}^{c \times m}$, where m is the number of representative local features. To determine the category of x^q, we need to calculate the similarity between its representative features and the prototype P of each specific category, where $\mathrm{P} = \left[\tilde{\mathrm{x}}_1, \tilde{\mathrm{x}}_2, ..., \tilde{\mathrm{x}}_{K*k_{psm}}\right] \in \mathbb{R}^{c \times (K*k_{psm})}$. However, directly using the local features of different semantic information may bring unreliable results. Therefore, we need to align the representative local features in the query example with those in the prototype, before computing their similarities.

In this work, the k-nearest neighbor algorithm is used to achieve semantic alignment. Specifically, when comparing a representative local feature x_i^q from a query image with the prototype of a category, we first use the similarity function $\phi(\cdot)$ calculate the similarity score between x_i^q and all local features in prototype, Then, we find the k-nearest neighbors $\tilde{\mathrm{x}}^j|_{j=1}^{k}$ of x_i^q in this prototype, and take the sum of k similarity scores with taking scores from TAM as weights. The weighted sum is the similarity between the x_i^q and prototype. Finally, we sum the similarity scores of all representative local features in a query example as its similarity score of the category corresponding to this prototype. Mathematically, the final similarity score $SScore$ is computed as:

$$SScore\ (query \rightarrow category) = \sum_{i=1}^{m} \sum_{j=1}^{k} score(\mathrm{x}_i^q) * \phi(\mathrm{x}_i^q, \tilde{\mathrm{x}}_i^j),$$

where $score(\mathrm{x}_i^q)$ is the score of x_i^q output by TAM and the similarity function $\phi(\cdot)$ is cosine similarity.

4. EXPERIMENTS

4.1. Dataset

We conduct experiments on two benchmark datasets: miniImageNet [8] and CUB [16]. The miniImageNet is a subset of the ImageNet dataset [17]. It consists of 100 classes, each of which contains 600 labeled images of size 84 × 84. We adopt the common setup introduced by [18], which defines a split of 64, 16 and 20 classes for training, validation and testing respectively. The CUB dataset contains 200 classes and 11,788 images in total. We used the splits from [19], where 100 classes are used for training, 50 for validation, and 50 for testing.

Table 1. Validation of the effectiveness of our proposed PSM and TAM. The result is the 5-way 5-shot mean accuracy (%) with a 95% confidence interval on the CUB (top) and miniImageNet (bottom) dataset.

Method	Backbone	Used Modules	5-way 5-shot
Baseline	Conv-64	-	80.83 ± 0.60
Baseline+PSM	Conv-64	+ PSM	82.94 ± 0.56
Baseline+PSM+TAM	Conv-64	+ PSM,TAM	**84.53 ± 0.65**
Baseline	ResNet-18	-	78.92 ± 0.66
Baseline+PSM	ResNet-18	+ PSM	80.13 ± 0.72
Baseline+PSM+TAM	ResNet-18	+ PSM,TAM	**81.21 ± 0.55**

Table 2. Time consuming comparison with MAML and ProtoNet on 5-way 5-shot setting.

Method	Backbone	training phase	test phase
ProtoNet	Conv-64	0.394s/iteration	0.264s/iteration
MAML	Conv-64	0.511s/iteration	0.301s/iteration
Our method	Conv-64	0.473s/iteration	0.281s/iteration

4.2. Training Details

We evaluate our method on 5-way 1-shot and 5-way 5-shot settings. Following the standard training strategy, we train 60,000 episodes in total for miniImageNet and 40,000 episodes for CUB. During the test phase, 600 test episodes are generated. We report the average accuracy as well as the corresponding 95% confidence interval over these 600 episodes. We consider Conv-64 [8], ResNet-18 [4] as our CNN-based embedding models for a fair comparison. The remaining parameters were selected based on the validation set.

4.3. Ablation Study

To better demonstrate the effectiveness of the proposed PSM and TAM, we develop a **baseline** for our method. Specifically, the prototype of each class is the average value of local features of all labeled images. During inference, we do not select representative local features for query examples but use all local features.

We first conduct experiments on CUB and miniImageNet with the backbone of Conv-64 and ResNet-18 respectively, with constantly adding PSM and TAM to the baseline method to see the effect of these two modules. As shown in Table 1, by comparing with our baseline, adding the PSM can obtain a 2.11% gain on CUB and 1.21% gain on miniImageNet. Adding the TAM, then the whole model is guided by our "task-specific guided" strategy, which can further improve the performance from 82.94% to 84.53% on CUB and 80.13% to 81.21% on miniImageNet. It indicates that unrelated background factors have side effects on performance, and our method can effectively reduce this interference by mining representative local features.

We further evaluate the computational complexity of our model by comparing it with two classical methods by testing the time consumption in each episode. From Table 2, Our method is efficient than MAML [20] which also needs "sub-training" as it requires second derivative to update model parameters, while achieves competitive time efficiency compared to ProtoNet [9]. Our method is developed based on ProtoNet. It is just mainly two more classifiers (each is a fully connected layer) than ProtoNet. One is trained to get the scaling factor ρ for each local feature in PSM, the other one in TAM is trained as a binary classifier to distinguish representative local features for query images. As both classifiers only need to train 5 epochs, therefore, our method is efficient both from theory and result.

Table 3. The mean accuracies (%) with a 95% confidence interval on the CUB dataset.

Method	Backbone	5-way 1-shot	5-way 5-shot
MAML [20]	Conv-64	55.92 ± 0.95	72.09 ± 0.76
Matching Net [8]	Conv-64	61.16 ± 0.89	72.86 ± 0.70
Prototypical Net [9]	Conv-64	51.31 ± 0.91	70.77 ± 0.69
RelationNet [10]	Conv-64	62.45 ± 0.98	76.11 ± 0.69
Baseline++ [19]	Conv-64	60.53 ± 0.83	79.34 ± 0.61
SAML [13]	Conv-64	69.33 ± 0.22	81.56 ± 0.15
DN4 [12]	Conv-64	53.15 ± 0.84	81.90 ± 0.60
Ours	Conv-64	**70.13 ± 0.62**	**84.53 ± 0.65**

Table 4. The mean accuracies (%) with a 95% confidence interval on the miniImageNet dataset. * means the confidence interval is not reported by the original work.

Method	Backbone	5-way 1-shot	5-way 5-shot
MAML [20]	Conv-64	48.70 ± 1.75	63.15 ± 0.91
Meta-SGD [21]	Conv-64	50.47 ± 1.87	64.03 ± 0.94
Reptile [22]	Conv-64	47.07 ± 0.26	62.74 ± 0.37
LEO [23]	WRN-28 [24]	61.76 ± 0.08	77.59 ± 0.12
Matching Net [8]	Conv-64	43.56 ± 0.84	55.31 ± 0.73
Prototypical Net [9]	Conv-64	49.42 ± 0.78	68.20 ± 0.66
RelationNet [10]	Conv-64	50.44 ± 0.82	65.32 ± 0.70
GNN [11]	Conv-64	50.33 ± 0.36	66.41 ± 0.63
Baseline++ [19]	Conv-64	48.24 ± 0.75	66.49 ± 0.63
SAML [13]	Conv-64	52.22 ± *	66.34 ± *
DN4 [12]	Conv-64	51.24 ± 0.74	71.02 ± 0.64
STANet-S [14]	Conv-64	53.11 ± 0.60	67.16 ± 0.66
CMT [15]	ResNet-18	62.05 ± 0.55	78.63 ± 0.06
FEAT [25]	Conv-64	55.15 ± *	71.61 ± *
Ours	Conv-64	53.98 ± 0.72	72.13 ± 0.63
Ours	ResNet-18	**62.79 ± 0.67**	**81.21 ± 0.55**

4.4. Comparison with State-of-the-art

We focus more on metric-based methods as our approach belongs to this kind. Based on comparison results on CUB and miniImageNet, which are shown in Table 3 and Table 4 respectively. Our method can achieve better or competitive performance compared to previous approaches. Especially DN4, SAML, STANet-S, and CMT, which also use local features, our method outperforms them by a sizable margin. Moreover, our method can achieve competitive accuracy to the recent FEAT with fewer parameters, as FEAT applies a more complicated Transformer [26] on the top of its backbone.

4.5. Conclusion

In this paper, we propose a simple and effective metric learning method based on local features to solve the few-shot learning problem. We propose a "task-specific guided" strategy to find local features that task-specific and discriminative according to the characteristics of each task. PSM and TAM are developed for support and query set respectively to support our strategy. Extensive experiments on the CUB, miniImageNet datasets verify the effectiveness of our method.

5. ACKNOWLEDGEMENT

This research was supported in part by the National Key R&D Program of China (2017YFB1200700), Capital Health Development Scientific Research Project (Grant 2020-1-4093), and Peking University Clinical Medicine + X Youth Project.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[6] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[7] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *ICML deep learning workshop*. Lille, 2015, vol. 2.

[8] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al., “Matching networks for one shot learning,” in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[9] Jake Snell, Kevin Swersky, and Richard Zemel, “Prototypical networks for few-shot learning,” in *Advances in neural information processing systems*, 2017, pp. 4077–4087.

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

[11] Victor Garcia Satorras and Joan Bruna Estrach, “Few-shot learning with graph neural networks,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[12] 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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 7260–7268.

[13] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao, “Collect and select: Semantic alignment metric learning for few-shot learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8460–8469.

[14] Shipeng Yan, Songyang Zhang, and Xuming He, “A dual attention network with semantic embedding for few-shot learning,” in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, 2019, pp. 9079–9086.

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

[16] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” Tech. Rep., 2011.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[18] Sachin Ravi and Hugo Larochelle, “Optimization as a model for few-shot learning,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[19] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang, “A closer look at few-shot classification,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[20] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[21] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li, “Meta-sgd: Learning to learn quickly for few-shot learning,” *arXiv preprint arXiv:1707.09835*, 2017.

[22] Alex Nichol, Joshua Achiam, and John Schulman, “On first-order meta-learning algorithms,” *arXiv preprint arXiv:1803.02999*, 2018.

[23] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell, “Meta-learning with latent embedding optimization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[24] Sergey Zagoruyko and Nikos Komodakis, “Wide residual networks,” in *Proceedings of the British Machine Vision Conference 2016, BMVC*, 2016.

[25] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha, “Few-shot learning via embedding adaptation with set-to-set functions,” in *2020 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2020*.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Proc. NIPS*, 2017, pp. 5998–6008.