Revisiting Metric Learning for Few-Shot Image Classification

Xiaomeng Li[1],Lequan Yu[1],Chi-Wing Fu[1], Meng Fang[3], Pheng-Ann Heng[1,2]

[1]*Department of Computer Science and Engineering, The Chinese University of Hong Kong*
[2] *Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China*
[3] *Tencent AI Lab*

Abstract

The goal of few-shot learning is to recognize new visual concepts with just a few amount of labeled samples in each class. Recent effective metric-based few-shot approaches employ neural networks to learn a feature similarity comparison between query and support examples. However, the importance of feature embedding, *i.e.*, exploring the relationship among training samples, is neglected. In this work, we present a simple yet powerful baseline for few-shot classification by emphasizing the importance of feature embedding. Specifically, we revisit the classical triplet network from deep metric learning, and extend it into a deep K-tuplet network for few-shot learning, utilizing the relationship among the input samples to learn a general representation learning via episode-training. Once trained, our network is able to extract discriminative features for unseen novel categories and can be seamlessly incorporated with a non-linear distance metric function to facilitate the few-shot classification. Our result on the miniImageNet benchmark outperforms other metric-based few-shot classification methods. More importantly, when evaluated on completely different datasets (Caltech-101, CUB-200, Stanford Dogs and Cars) using the model trained with miniImageNet, our method significantly outperforms prior methods, demonstrating its superior capability to generalize to unseen classes.

Keywords: Few-shot learning, metric learning, feature representation, deep learning.

1. Introduction

Learning from a few data is a hallmark of human intelligence, however, it remains a challenge for modern deep learning systems. Recently, there has been a growing interest in few-shot learning [1–26], which aims to recognize new visual concepts with just a small amount of labeled data for training. In other words, the goal of few-shot learning is to classify unseen data instances (query examples) into a set of new categories, given just a small number of labeled instances in each class (support examples). In this work, we focus on the case of few-shot classification, where *only a few labeled examples per class* are given.

Obviously, naively fine-tuning a model on the novel labeled data would easily overfit the *few* given data. Hence, data augmentation and regularization [27, 28] are often employed to somehow relieve the overfitting. Later, the meta-learning paradigm [3, 4, 29, 30] shed light to the few-shot learning problem; several metric learning-based methods [2, 31–33] were developed. For instance, the *matching network* [32] uses an end-to-end trainable k-nearest neighbors algorithm on the learned embedding of the few labeled examples (*support set*) to predict the classes of the unlabeled samples (*query set*), while the *prototypical network* [2] further builds a pre-class prototype representation. More recently, Sung et al. presented the *relation network* [33], which learns a nonlinear distance metric via a shallow neural network instead of using a fixed linear distance metric, eg, Cosine [32] and Euclidean [2]. These methods used sampled mini-batches called *episode* to train an end-to-end network, aiming at making the training process more faithful to the test environment. Although these methods utilize deep networks to extract expressive deep features, they do not take full advantages of the relationship among the input samples. Hence, we are motivated to explore strategies to improve the feature embedding in terms of their efficiency to be transferable to handle unseen class samples and their generality for few-shot classification. Although the triplet-like feature embedding is a longstanding topic in the computer vision area, its importance and effectiveness for the few-shot classification is neglected by the community.

In this work, we revisit metric learning and investigate the potential of triplet-like feature embedding learning for few-shot classification. We aim to *meta-learn a feature embedding* that performs well, not only on the training classes but more importantly, on the novel classes. Specifically, the feature embedding should map the similar samples close to one another and dissimilar ones far apart. This is well-aligned with the philosophy of triplet-like learning. However, the general triplet network only interacts with a single negative sample per update, while few-shot classification requires a comparison with multiple query samples, typically of different classes. Hence, we formulate an improved triplet-like metric learning, namely the *deep K-tuplet Network*, to improve the few-shot classification. Particularly, the *deep K-tuplet Network* generalizes the triplet network to allow joint comparison with K negative samples in each mini-batch. It makes the feature embedding learning process more faithful to the few-shot classification problem with improved feature generalization. Moreover, we present the *semi-hard mining* sampling technique, an effective sampling strategy to sample informative hard triplets. Hence, we can speed up the convergence and stabilize the training procedure.

Our technique is simple yet powerful, and can be seamlessly incorporated with the learnable non-linear distance metric [33] for few-shot classification. To demonstrate the generalization capability of our presented few-shot classification framework, we train our model on the miniImageNet dataset [32], and conduct few-shot classification, not only on the miniImageNet testing data, but also on other novel classes in other datasets (e.g., Caltech-101, CUB-200, Stanford Dogs and Cars). Experimental results demonstrate that our method effectively generalizes for unseen novel class samples, even across different datasets.

The main contributions of this work are threefold:

1. We present a simple and powerful baseline method to investigate the importance of feature embedding for few-shot classification, where the effectiveness of feature embedding is neglected by previous works.
2. We present the *deep K-tuplet Network* to effectively learn the discriminative feature embedding on unseen class samples for few-shot learning. Our method outperforms other metric-based methods and achieves competitive performance over other meta-based methods on the miniImageNet.
3. More importantly, prior works evaluated the few-shot learning within one dataset, *i.e.*, the novel classes and base classes are sampled from the same dataset. This experiment setting may be not representative in the real world setting. We establish a new experimental setting for evaluating the cross-domain generalization ability for few-shot classification algorithms. Our result generalized on CUB-200, Stanford Dogs, Stanford Cars and Caltech-101 excels other methods, showing the excellent cross-domain generalization capacity of our method.

2. Related Work

Few-shot learning is an important area of research. Early works on the few-shot learning focused on generative models and inference strategies [34, 35]. In [34], the authors assumed that one can utilize knowledge coming from previously-learned classes to make predictions on new classes only with one or few labels. However, these methods do not involve deep learning. Recently, with the success of deep learning, significant progress has been achieved in the few-shot learning area.

2.1. Meta-learners for Few-Shot Learning

One category of the few-shot learning is meta-learner based methods [3, 4, 29, 30, 36]. The meta-learning algorithm (MAML) [3] used a model agnostic meta-learner to train a good basic model on a variety of training tasks, such that given a new task with only a few training samples, a small number of gradient steps is sufficient to produce a good generalization model. Ravi & Larochelle [4] further proposed an LSTM-based meta-learning model to learn the optimization algorithm of training a network, where the LSTM updates the weights of a classifier for a given episode. Both methods, however, need to fine-tune the basic model on the target problem. Munkhdalai & Yu [30] introduced a novel meta-learning architecture that learns meta-level knowledge across tasks and produces a new model via fast parameterization for rapid generalization. Santoro *et al.* [29] introduced a memory-augmented neural network to quickly encode and retrieve new data and make accurate predictions with only a few samples. Lately, some other works [37–39] focused on meta-learners for few-shot classification. However, all these methods need to fine-tune or update the parameters for new unseen tasks, while our method performs the target tasks based entirely on feed-forward without requiring further parameter updates.

2.2. Deep Metric Learning

Our work is related to deep metric learning, which involves a large volume of metric learning methods [40–44]. Below, we briefly review the more relevant ones. The

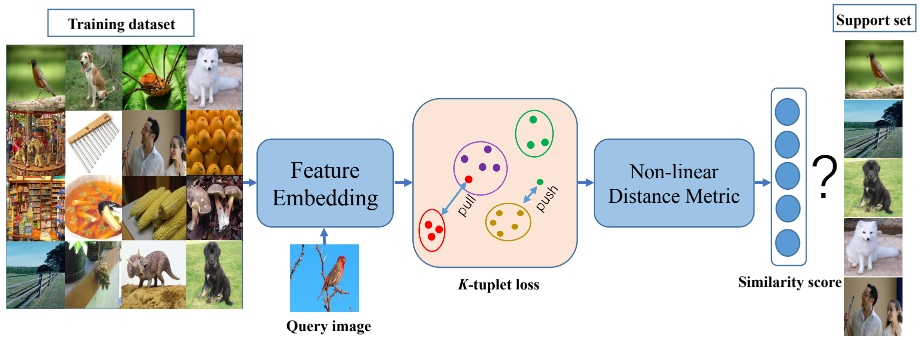

Figure 1: The framework of *deep K-tuplet Network* for few-shot learning. We train a embedding network to learn transferable feature embedding via the K-tuplet loss from the training dataset. The anchor interacts with multiple negative images in the tuplet, and contributes to the discriminative features. The well-learned embedding features are fed into the non-linear distance metric module to learn similarity among the query image and samples in the support set. Finally, we perform few-shot classification on the novel category.

goal of metric learning is to minimize intra-class variations and maximize inter-class variations. Early works used the siamese architecture [41, 45] to capture the similarity between images. The recent works [42, 46, 47] adopted the deep networks as the feature embedding function and used triplet losses instead of pairwise constraints to learn the metric. These metric learning strategies have been widely used in image retrieval [46], face recognition [47–50] and person re-identification [50, 51]. For example, Lu *et al.* [49] proposed a discriminative deep metric learning method for face and kinship verification, where the distance of each positive pair is reduced and that of each negative pair is enlarged. Hu *et al.* [50] proposed a multi-view metric learning (MvML) to jointly learn an optimal combination of multiple distance metrics on multi-view representation. It learns a shared representation for different views and the method is applied on face verification, kinship verification, and person re-identification. Duan *et al.* [52] presented a deep adversarial metric learning (DAML)to generate synthetic hard negatives from the observed negative samples, where the potential hard negatives are generated to the learned metric as complements. More recently, Wu *et al.* [53] presented a feature embedding method based on neighborhood component analysis. These works show that combining deep models with proper objectives is effective in learning the similarities. Unlike these methods, we consider using triplet-like networks to improve the feature discrimination on the unseen class images for few-shot learning problem.

2.3. Metric Learning for Few-shot Learning

The second branch are metric based approaches [2, 31–33, 53–56]. Metric learning based methods learn a set of project functions (embedding functions) and metrics to measure the similarity between the query and samples images and classify them in a feed-forward manner. The key difference among metric-learning-based methods lies in how they learn the metric. Koch *et al.* [31] presented the siamese neural networks to compute the pair-wise distance between samples, and used the learned distance to solve the one-shot learning problem via a K-nearest neighbors classification. Vinyals *et* *al.* [32] designed end-to-end trainable k-nearest neighbors using the cosine distance on the learned embedding feature, namely matching network. Lately, Snell *et al.* [2] extended the matching network by using the Euclidean distance instead of the cosine distance and building a prototype representation of each class for the few-shot learning scenario, namely prototypical network. Mehrotra & Dukkipati [57] trained a deep residual network together with a generative model to approximate the expressive pair-wise similarity between samples.

Recently, Ren *et al.* [58] extended the prototypical network to do semi-supervised few-shot classification, while Garcia *et al.* [55] defined a graph neural network to conduct semi-supervised and active learning. Sung *et al.* [33] argued that the embedding space should be classified by a nonlinear classifier and designed the *relation module* to learn the distance between the embedded features of support images and query images. The relation network extends the matching network and prototypical network by including a learnable nonlinear comparator. Notably, the prototypical networks [2], siamese networks [31], and relation net [33] all adopt the episode-based training strategy, where each episode is designed to mimic few-shot learning. More recently, Li *et al.* [59] proposed category traversal module (CTM) to look at all categories in the support set to find task-relevant features. Li *et al.* [6] present the deep nearest neighbor neural network to improve the final classification in the few-shot learning. Although the excellent performance achieved in the few-shot classification, the importance of feature embedding has not paid sufficient attention.

3. Method

3.1. Overview

Few-shot classification involves three datasets: a training set $\mathcal{D}_{train}$, a support set $\mathcal{D}_{supp}$, and a query set $\mathcal{D}_{query}$. In short, we want to train a model to learn transferable knowledge from $\mathcal{D}_{train}$, and apply the knowledge in the testing phase to classify the samples in $\mathcal{D}_{query}$ given $\mathcal{D}_{supp}$.

- $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{N}$ is used for training the model, where x_i is a training image, $y_i \in C_{train}$ is the label of x_i, and N is the number of training examples.
- $\mathcal{D}_{supp} = \left\{(x_j, y_j)\right\}_{j=1}^{M}$ is the set of M labeled examples given in the testing phase, where $y_j \in C_{supp}$ is the label of x_j but $C_{train} \cap C_{supp} = \varnothing$.
- Given $\mathcal{D}_{query} = \left\{x_j\right\}_{j=1}^{n}$, the goal of few-shot classification is to classify the samples in $\mathcal{D}_{query}$.

Note that the support set $\mathcal{D}_{supp}$ and the query set $\mathcal{D}_{query}$ share the same label space. If the support set has K labeled examples for each of the C classes in C_{supp}, *i.e.*, $M = C \times K$, then the few-shot problem is called C-way K-shot.

Figure 1 overviews our few-shot learning framework. First, we meta-learn a transferable feature embedding through the *deep K-tuplet network* with the designed K-tuplet loss from the training dataset. The well-learned embedding features of the query image and samples in the support set are then fed into the non-linear distance metric to learn the similarity scores. Further, we conduct few-shot classification based on these scores.

3.2. Meta-learn Feature Embedding

Such nonlinear mapping should be generalizable to work with samples of novel classes, meaning that the mapping should preserve the class relationship on the unseen class samples in $\mathcal{D}_{supp}$ and $\mathcal{D}_{query}$. We adopt a *triplet-like network* to learn the feature embedding on $\mathcal{D}_{train}$.

Specifically, for an input image x_i, function $f(\cdot;\theta) : \mathcal{X} \to \mathbb{R}^d$ maps x_i to an embedding vector $f(x_i)$, where θ denotes the parameters of the embedding function; d is the dimension of the embedded features, and $f(x_i)$ is usually normalized to unit length for training stability and comparison simplicity. To learn parameter θ, the traditional triplet loss is widely used, where the objective is based on a relative similarity or distance comparison metric on the sampled pairs. In short, the training samples are randomly selected to form a triplet (x_a, x_p, x_n) with an anchor sample x_a, a positive sample x_p, and a negative sample x_n. The label of the selected samples in a triplet should satisfy $y_a = y_p \neq y_n$. The aim of the loss is to pull $f(x_a)$ and $f(x_p)$ close to each other, while pushing $f(x_a)$ and $f(x_n)$ far apart.

However, the above traditional triplet loss interacts with only one negative sample (and equivalently one negative class) for each update in the network, while we actually need to compare the query image with multiple different classes in few-shot classification. Hence, the triplet loss may not be effective for the feature embedding learning, particularly when we have several classes to handle in the few-shot classification setting. Inspired by [43], we generalize the traditional triplet loss to a tuplet loss with K-negatives, namely K-tuplet loss, to allow simultaneous comparison jointly with K negative samples, instead of just one negative sample, in one mini-batch. This extension makes the feature comparison more effective and faithful to the few-shot learning procedure, since each update, the network can compare a sample with multiple negative classes altogether.

In particular, we randomly choose the K negative samples $x_{n_i}, i = \{1, 2, .., K\}$ to form into a triplet. Accordingly, the optimization objective is formulated as:

$$\mathcal{L}(x_a, x_p, x_{n_i}) = \frac{1}{K} \sum_{i \in U} \left[\left\| f_a - f_p \right\|^2 - \left\| f_a - f_{n_i} \right\|^2 + \alpha \right]_+, K = |U| \tag{1}$$

where $[\cdot]_+ = \max(0, \cdot)$ denotes the hinge loss function, α is the hyperparameter margin and U denotes the set of triplets, and we write $f(x)$ as f to omit x for simplicity. For the anchor sample x_a, the optimization shall maximize the distance to the negative samples x_{n_i} to be larger than the distance to the positive sample x_p in the feature space. To form one mini-batch to train the network, we randomly select B anchor samples from the training set, where B is batch size. For each anchor sample x_a, we then randomly select another positive sample x_p of the same class as x_a and further randomly select K other negative samples whose classes are different from x_a. Among the K negative samples, their class labels may be different. Figure 2 visualizes the K-tuplet loss and triplet loss. When K equals to 1, K-tuplet loss becomes triplet loss. The classification accuracy is improved with a larger K,since the anchor sample interacts

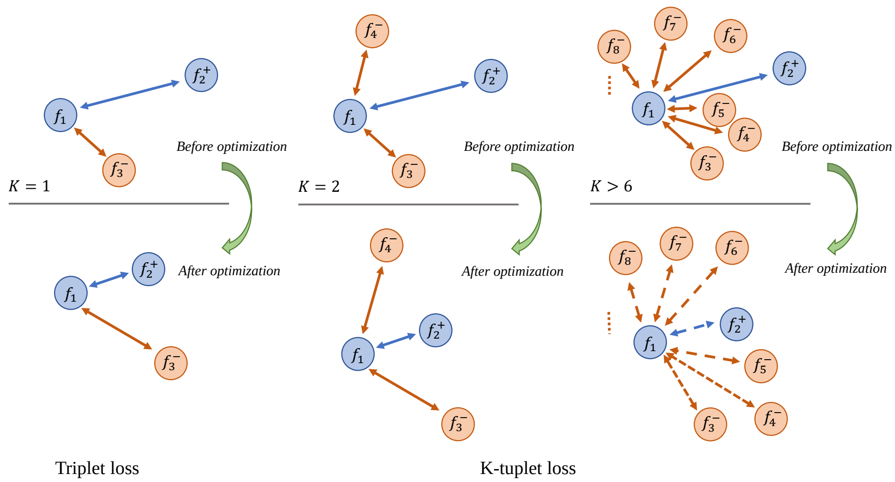

Figure 2: The visualization of K-tuplet loss function. When K equals to 1, K-tuplet loss becomes triplet loss.

with more samples in one mini-batch and makes the gradient more stable. However, when K increases, the computational burden increases and it becomes too heavy-lifting to perform standard optimization, i.e., stochastic gradient descent (SGD) with a mini-batch. In other words, the classification accuracy would decrease as K continues to increase. From our experiments, we can achieve the best performance when setting K to 5. Compared with the traditional triplet loss, more inter-class variations have been considered in each forward update by using our K-tuplet loss, thus making the learned feature embedding more discriminative for samples from different classes.

3.3. Efficient Training with Semi-hard Mining

The semi-hard mining strategy is motivated by the observation that when the model starts to converge, the "well-learned easy samples" will obey the margin and could not contribute to the optimization in the learning process. However, the "hard samples" still fail to satisfy the optimization goal. This phenomenon degrades the model performance and also slows down the convergence of the training. We, thereby, design a semi-hard mining strategy to sample more informative hard triplets in each mini-batch when the model starts to converge. The informative hard triplets are selected by whether the condition in the loss function is satisfied or not. The loss function of semi-hard mining can be described as the following:

$$\begin{gathered}\mathcal{L}_{semi-hard}(x_a, x_p, x_{n_i}) = \frac{1}{s}\sum_{i \in S}\left[\left\|f_a - f_p\right\|^2 - \left\|f_a - f_{n_i}\right\|^2 + \alpha\right]_+, \\ \text{where } S = \{i \in U \mid \|f_a - f_{n_i}\|^2 - \|f_a - f_p\|^2 \geq \alpha\} \text{ and } s = |S|\end{gathered} \tag{2}$$

where $[\cdot]_+ = \max(0, \cdot)$ denotes the hinge loss function and α is hyperparameter margin. x_a, x_p, x_n denote an anchor, positive and negative sample, respectively. s is the number of elements in set S, where set S represents the triplets that are selected as informative and hard. We write $f(x_a)$ as f_a to omit x for simplicity.

This semi-hard loss function is utilized when the model starts to converge (80 epochs in our experiments) and we continue to fine tune it for 100 epochs. We utilize Adam optimizer with a learning rate of 0.001 to train the network. We analyze the effectiveness of this technique in Table 5. From our experiments, we can see that this semi-hard mining strategy helps improve the training efficiency and contributes to the learning of feature embedding.

3.4. Non-linear Distance Metric Learning

Furthermore, we adopt the non-linear distance metric module [33] to learn to compare the embedded features in few-shot classification. Given image x_s from the support set and image x_q from the query set, their similarity score is learned by concatenating $f_\theta(x_q)$ and $f_\theta(x_s)$ and then feeding the combined feature into a non-linear distance metric. The non-linear distance metric has two convolutional blocks and two fully-connected layers. Each convolutional block consists of a 3×3 convolution with 64 channels followed by a batch normalization, an ReLU activation function, and a 2×2 max-pooling. The fully-connected layers have 8 and 1 outputs, followed by a sigmoid function to get the final similarity scores between the query image x_q and samples in the support set. In the end, our non-linear distance metric learns to produce the similarity score by calculating the mean square error loss, following the same spirit as [33].

Figure 3 shows the detailed network architecture of our nonlinear metric learning module. The input is the concatenation of features from the images of the support set and the query set. The output is the similarity scores of the query images with images in the support set. The few-shot classification prediction is the label of the image that has the maximum similarity score in the support set.

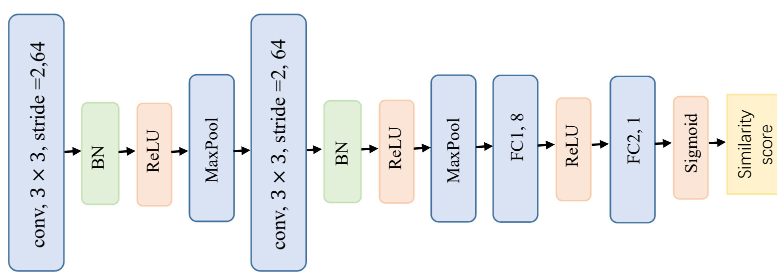

Figure 3: The detailed network architecture of our nonlinear metric learning module. The last number in each box denotes the number of feature channels.

3.5. Technique Details

We employed the ResNet34 architecture [60] for learning the feature embedding. When meta-learning the transferable feature embedding, we used Adam optimizer [61] with a learning rate of 0.001 and a decay for every 40 epochs. We totally trained 100 epochs and adopted the semi-hard mining strategy when the loss starts to converge (at around 80 epoches). To learn the non-linear distance metric, we followed the episode-based strategy and also employed the Adam optimizer with a learning rate of 0.001. Different from the general episode sampling procedure, we sampled multiple episodes to form each mini-batch to train the non-linear distance metric. This strategy increases the data diversity, *i.e.*, the number of different class samples) and makes the training more stable.

We evaluate the accuracy of few-shot classification by averaging the randomly-generated episodes from the training set, following [2]. For 5-way 1-shot test, each query image is compared with five samples in the support set. The prediction is the label of the sample that has the maximum similarity score within the support set. For 5-way 5-shot test, we sum the features of all the samples in each class in the support set as the feature map of the class and then follow the same procedure with 5-way 1-shot setting to get the query image label.

4. Experiments

We first evaluate our few-shot classification method on the public miniImageNet dataset. We then show the generalization of our approach by directly evaluating on completely different datasets using the model trained with miniImageNet. Lastly, we extensively analyze the different components of our method.

4.1. Few-shot Classification on the miniImageNet

The miniImageNet dataset is derived from the ILSVRC-12 dataset [62], consisting of 60,000 color images with 100 classes and 600 samples per class. In order to directly compare with state-of-the-art algorithms, we follow the splits introduced by Ravi and Larochelle [4], with 64, 16 and 20 classes for training, validation and testing, respectively. The validation dataset is used for monitoring generalization performance of the network only and not used for training the network.

We compare our approaches with several state-of-the-art methods reported on the miniImageNet [2, 32, 33], as shown in Table 1. Most of the existing methods employed the shallow neural network, *i.e.*, four convolutional layers, to extract the feature. Since our method is based on the well-learned feature embedding, the shallow embedding network did not make adequate usage of our method's expressive capacity. Thus, we follow the recent works [11, 37, 63] to use a deeper embedding network, *i.e.*, ResNet, to prevent the underfitting.

Compared with metric-based methods, we can see that our method achieves the highest accuracy on 5-way 1-shot setting and very competitive accuracy on 5-way 5-shot setting, as shown in Table 1. Note that Li *et al.* [6] achieves 54.37 ± 0.36 % and 74.44 ± 0.29 % with ResNet backbone on 5-way 1-shot and 5-way 5-shot respectively. However, our result outperforms their method on 5-way 1-shot and shows competitive result on 5-way 5-shot setting. We report the few-shot classification accuracy of our method using the K-NN classifier with the Euclidean distance on the embedded feature; see *Ours+Euclid* in Table 1. In this setting, we remove the non-linear metric and use K nearest neighbors ($K = 1$) on the embedded features of query images and support images for classification. It is observed that the Euclid version of our method still achieves the competitive results, showing the generalization and discrimination of the learned feature embedding on unseen novel categories.

Beside metric-based methods, there are several state-of-the-art meta-learning approaches for the few-shot learning problem [11, 36, 38, 39]. For example, Gidaris *et al.* [38] propose a dynamic-net, and report 56.20 ± 0.86 (%) on 5-way 1-shot and 72.81 ± 0.62 (%) on 5-way 5-shot setting. Andrei A *et al.* [36] proposed a latent embedding optimization (LEO) meta-learning approach that decouples the gradient-based adaptation procedure from the underlying high-dimensional space of model parameters. Their method achieved 61.76 ± 0.08 (%) on 5-way 1-shot settings. More recently, Kwonjoon *et al.* [11] explored two properties of linear classifiers in meta-learning, *i.e.*, implicit differentiation of the optimality conditions of the convex problem and the dual formulation of the optimization problem. Their method achieved 62.64 ± 0.61 (%) on 5-way 1-shot and 78.63 ± 0.46 (%) on 5-way 5-shot setting. However, this method is based on the ResNet 12 backbone and the direct comparison is not fair. As our work learns deep metrics in the embedding space, we mainly compare with metric-based approaches. More importantly, our method has only one single unified network, which is much simpler than these meta-learning-based methods with additional complicated memory-addressing architectures.

In Figure 4, we show the 10 nearest neighbor images of the query image on the miniImageNet testing dataset with the Euclid distance of our learned embedding features. We can see our feature embedding preserves apparent visual similarity better and facilitates the accurate recognition.

Table 1: Average few-shot classification accuracies (%) on the miniImageNet. Note that '-' denotes not reported. All accuracy results are averaged over 600 test eposides and are reported with 95% confidence intervals.

Model	Year	5-way Acc.	
		1-shot	5-shot
Matching Nets [32]	2016 NIPS	46.6 ± 0.8	60.0 ± 0.7
Meta-Learn LSTM [4]	2017 ICLR	43.44 ± 0.77	60.60 ± 0.71
MAML [3]	2017 ICML	48.70 ± 1.84	63.11 ± 0.92
Meta Nets [30]	2017 ICML	49.21 ± 0.96	-
Proto Net [2]	2017 NIPS	49.42 ± 0.78	68.20 ± 0.66
Proto Net (*ResNet*) [2]	2017 NIPS	51.15 ± 0.85	69.02 ± 0.75
Triplet ranking [64]	2018 Arxiv	48.76	-
GNN [55]	2018 ICLR	50.33 ± 0.36	66.41 ± 0.63
Masked Soft k-Means [58]	2018 ICLR	50.41 ± 0.31	64.39 ± 0.24
Relation Net [33]	2018 CVPR	50.44 ± 0.82	65.32 ± 0.70
Relation Net (*ResNet*) [33]	2018 CVPR	52.13 ± 0.82	64.72 ± 0.72
large margin few-shot [65]	2018 Arxiv	51.08 ± 0.69	67.57 ± 0.66
SNAIL [37]	2018 ICLR	55.71 ± 0.99	68.88 ± 0.92
R2D2 [8]	2019 ICLR	51.2 ± 0.6	68.8 ± 0.1
DN4 [6]	2019 CVPR	51.24 ± 0.74	71.02 ± 0.64
Ours+Euclid	-	54.46 ± 0.89	68.15 ± 0.65
Ours	-	**58.30 ± 0.84**	**72.37 ± 0.63**

4.2. Generalizing to Other Datasets

A new dataset may present data distribution shift, and the classification accuracy of widely used models drops significantly [67]. In current setting of few-shot classifi-

Figure 4: Nearest neighbors from the learned feature embedding of our method on the miniImageNet testing dataset. Given a query image, we shows 10 nearest neighbor images.

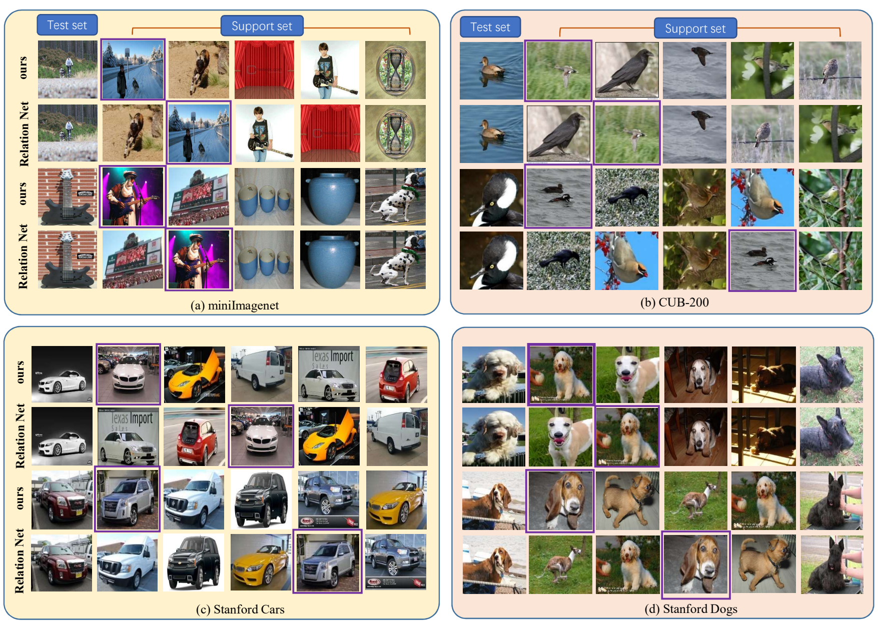

Figure 5: Examples of visualized results of few-shot classification on (a) miniImageNet, (b) CUB-200, (c) Stanford Cars and (d) Stanford Dogs dataset. The images in the support set is sorted by the similarity with the test image (from left to right and only showing top-5 images). Purple box denotes the ground-truth class in the support set.

Table 2: Average few-shot classification accuracies (%) on other datasets using the models trained with the miniImageNet. Note that all the experiments are conducted with the same network for fair comparison.

Dataset		Proto Net [2]	Relation Net [33]	Cosface embed [66]	Ours
Caltech-101	5-way 1-shot	53.28 ± 0.78	53.50 ± 0.88	57.22 ± 0.85	**61.00 ± 0.81**
	5-way 5-shot	72.96 ± 0.67	70.00 ± 0.68	75.34 ± 0.69	**75.60 ± 0.66**
CUB-200	5-way 1-shot	39.39 ± 0.68	39.30 ± 0.66	39.60 ± 0.70	**40.16 ± 0.68**
	5-way 5-shot	56.06 ± 0.66	53.44 ± 0.64	55.70 ± 0.66	**56.96 ± 0.65**
Stanford Dogs	5-way 1-shot	33.11 ± 0.64	31.59 ± 0.65	43.16 ± 0.84	**37.33 ± 0.65**
	5-way 5-shot	45.94 ± 0.65	41.95 ± 0.62	49.32 ± 0.77	**49.97 ± 0.66**
Stanford Cars	5-way 1-shot	29.10 ± 0.75	28.46 ± 0.56	29.57 ± 0.70	**31.20 ± 0.58**
	5-way 5-shot	38.12 ± 0.60	39.88 ± 0.63	40.78 ± 0.68	**47.10 ± 0.62**

cation, most methods conduct training and testing phases within the same dataset, *i.e.*, miniImageNet. Although the training classes and testing classes do not share the same label space, they still comes from the same data distribution. While, in the real world, the unknown novel classes may comes from an agnostic data distribution. Therefore, to validate the generalization capability of our approach, we conduct the few-shot classification on novel classes from the following four datasets using the model trained on the miniImageNet training dataset.

- **Caltech-101.** The Caltech-101 dataset [34, 68] contains objects belonging to 101 categories. Each category contains about 40 to 800 images. Most categories have about 50 images.
- **Caltech-UCSD Birds-200-2011 (CUB-200).** Caltech-UCSD Birds 200 (CUB-200) [69] contains photos of 200 bird species (mostly North American). In this fine-grained dataset, subtle differences between very similar classes can hardly be recognized even by humans.
- **Stanford Dogs.** The Stanford Dogs dataset [70] contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization.
- **Stanford Cars.** The Stanford Cars [71] contains 16,185 images of 196 classes of cars.

Following the same data selection principal as miniImageNet [32], we randomly select 20 classes in each dataset as the test dataset. Note that the test datasets do not share the same label space with the training images. Please see the section 1 in the supplementary files for detailed selected class in each dataset. Without any fine-tuning, we directly use the model trained on the miniImageNet training dataset to perform few-shot classification on the new datasets. Table 2 shows the classification performance of Relation Net, Proto Net and our method on the four datasets. The results are achieved by the model with the same network backbone. It is observed that our model performs consistently better than Relation Net and Proto Net on all four datasets. To compare the results on the different datasets, the accuracy on Caltech-101 are much higher than the results of other three datasets, even than the miniImageNet testing dataset. This is because the Caltech-101 contains a single object with pure background and it is much easier to be recognized, while the CUB-200, Stanford Dogs and Stanford Cars have relative complex background. We visualize the results of 5-way 5-shot setting achieved by Relation Net and our model in Figure 5. We can see that our method is very discriminative to similar objects. These comparisons clearly demonstrate that our approach is able to learn more generalized transferable features for few-shot classification among different datasets. Please see more visualized results in the section 3 in the supplementary files.

4.3. Analysis of Our Method

To better understand our method, we conduct the following experiments on the miniImageNet dataset.

4.3.1. Results with Different Network Backbones

We compare the few-shot classification performance of our approach under different network backbones, *i.e.*, AlexNet [72], VGG [73] and ResNet [60]. We conduct experiments on the miniImageNet with the same experiment setting for network architectures. Note that the performance is evaluated by one nearest neighborhood (1-NN) with Euclidean distance to better show the influence of different network backbones. From the results in Table 3, it is observed that the classification accuracy of AlexNet and VGG11 are similar, while the few-shot classification accuracy is largely improved (about 10% improvement in both 5-way 1-shot and 5-way 5-shot settings) with a more deeper ResNet. The reason may be that we can extract more representative features with the deeper ResNet and thus improve the accuracy on few-shot testing. ResNet18 and ResNet34 achieve similar results on 5-way 5-shot evaluation, but ResNet34 achieves a bit higher performance on 5-way 1-shot setting. However, the classification accuracy would be decreased as the model complexity continues to grow (*e.g.*, from ResNet34 to ResNet50). This finding indicates that too many parameters may lead to overfitting on the training tasks and thus decrease the classification results on novel categories. Therefore, an effective network backbone can indeed contribute to the transferable feature extraction and improve the accuracy on few-shot classification. Overall, in our experiment, we choose the ResNet34 as the network backbone.

Table 3: Few-shot classification accuracy (%) for 600 runs with 95% confidence intervals with different network backbones on the miniImageNet testing data.

Backbone	5-way Acc.	
	1-shot	5-shot
AlexNet	44.78 ± 0.78	58.69 ± 0.71
VGG 11	44.74 ± 0.81	58.55 ± 0.66
ResNet18	53.62 ± 0.84	**68.27 ± 0.67**
ResNet34	**54.46 ± 0.89**	68.15 ± 0.65
ResNet50	53.46 ± 0.88	65.32 ± 0.72

4.3.2. The Tuplet-loss with Different Negative Pairs

We compare the performance of our method with different K in the tuplet loss, where K is the number of negative samples from different classes in each tuplet. We

Table 4: Few-shot classification accuracy (%) for 600 runs with 95% confidence intervals on the miniImageNet testing data with different number of negatives in the tuplet loss .

Number of K	5-way Acc.	
	1-shot	5-shot
K=1 (*triplet loss*)	40.15 ± 0.75	54.62 ± 0.68
K=4	51.22 ± 0.81	65.66 ± 0.68
K=5	**54.46 ± 0.89**	**68.15 ± 0.65**
K=8	53.17 ± 0.81	66.77 ± 0.68
K=16	46.03 ± 0.79	60.02 ± 0.67

also report the classification accuracy using the one nearest neighborhood (1-NN) classifier with Euclidean distance. As shown in Table 4, the accuracy is a little low if we set K as 1 in the tuplet loss (equivalent to traditional triplet loss). The classification accuracy is improved with a larger K, since the anchor sample interacts with more samples in one mini-batch and makes the gradient more stable. In another aspect, the classification accuracy would be saturated with a bigger K, and we can achieve the best performance when setting K to 5.

It is worth mentioning that when K equal to 1, K-tuple loss is triplet loss. Compared with triplet loss, the classification accuracy is improved with K-tuplet loss, since the anchor sample interacts with more samples in one mini-batch and makes the gradient more stable. As shown in firs line in Table 4, the performance of triplet loss (K =1) on our few-shot learning task is 40.15% and 54.62% for 1-shot and 5-shot learning. While our K-tuplet loss can achieve 54.46% and 68.15% for 1-shot and 5-shot learning respectively, surpassing the triplet loss by around 14% on both 1-shot and 5-shot setting.

4.3.3. Effects of Semi-hard Mining

Table 5 shows the effects on our feature embedding when trained with and without semi-hard mining. We report the few-shot classification accuracy on the *mini*ImageNet testing data with the two resulting learned feature embedding. “w/o semi-hard” denotes the model trained with equation (1) for 100 epochs while “w semi-hard” refers to model trained with equation 1 for 80 epochs and then utilize equation (2) for remaining 20 epochs. It is observed that with semi-hard mining, the few-shot classification accuracy on both 1-shot and 5-shot scenarios can be further improved by relative 1.6% and 1.0% respectively. This comparison demonstrates the effectiveness of “semi-hard mining strategy” to improve feature embedding learning.

Table 5: The effects of semi-hard mining. The report results are averaged few-shot classification for 600 runs with 95% confidence intervals (Unit: %).

Setting	5-way acc	
	1-shot	5-shot
w/o semi-hard	53.62 ± 0.84	67.48 ± 0.68
w semi-hard	**54.46 ± 0.89**	**68.15 ± 0.65**

Table 6: Few-shot classification accuracy (%) for 600 runs with 95% confidence intervals on the miniImageNet testing data with different margins.

Margin α	5-way Acc.	
	1-shot	5-shot
0.10	54.46 ± 0.89	68.15 ± 0.65
0.30	57.10 ± 0.64	71.02 ± 0.58
0.50	**58.30 ± 0.84**	**72.37 ± 0.63**
0.80	53.05 ± 0.41	62.02 ± 0.51
1.00	49.30 ± 0.07	59.62 ± 0.07

Table 7: Averaged accuracy for 600 runs with 95% confidence intervals on the miniImageNet testing data with additional training classes from ImageNet dataset. N is the the number of extra classes in the notation of "64+N".

Training Class	5-way Acc.	
	1-shot	5-shot
64 + 0	58.30 ± 0.84	72.37 ± 0.63
64 + 64	61.12 ± 0.06	75.14 ± 0.67
64 + 128	**65.60 ± 0.07**	**77.74 ± 0.07**

4.3.4. The Analysis of Different Margin

We also investigate the effect of different margin α in the tuplet loss and the results of our whole framework with different settings are shown in Table 6. The experimental results show that with margin 0.5, the feature embedding in this task is the best. A smaller margin will decrease the performance due to the inter-class variation is not well-learned. And a larger margin may increase the difficulty in the training.

4.3.5. The Results with More Training Classes

We would like to explore whether the few-shot classification accuracy will increase if more training classes are available. Thus, we conduct experiments with additional class images from the ImageNet dataset. Note that the additional dataset does not share the same labels with the testing images. Table 7 presents the accuracy on the miniImageNet testing dataset of our method trained with different number of training classes. We can see that our method can be further improved with extra training classes data available. This is conform with our expectation that we can learn more transferable generalized feature embedding from more training samples. Based on the generalized feature, we can further improve the few-shot classification accuracy on the novel categories.

5. Comparison of Visualized Features

The effectiveness of our method is mainly due to a well-learned feature embedding, which improves the few-shot classification performance on the *novel classes*. To show the generalization and discrimination of our learned feature embedding on novel class samples, we visualize the features in comparison with Proto Net [2] and Relation

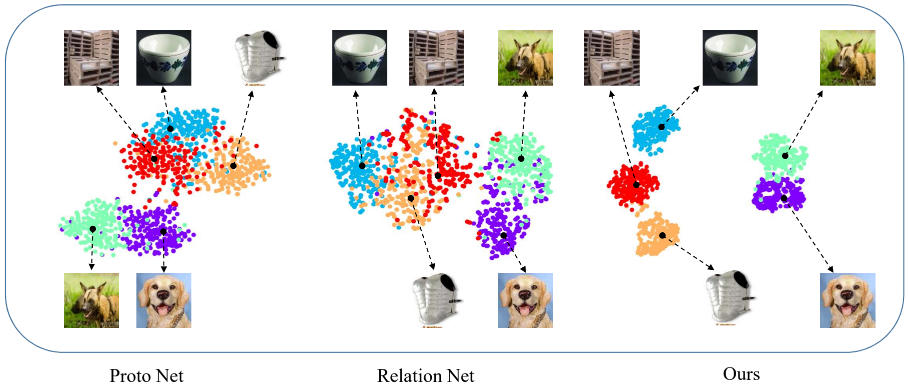

Figure 6: T-SNE visualization of features in Proto Net, Relation Net and our method on the same set of samples in the test dataset (example 1).

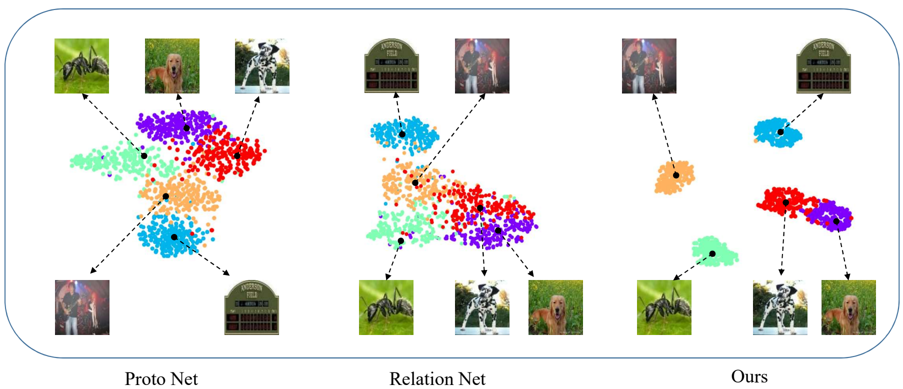

Figure 7: T-SNE visualization of features in Proto Net, Relation Net and our method on the same set of samples in the test dataset (example 2).

Net [33]; see Figures 6 and 7. The feature embedding is learned from the miniImageNet training dataset and tested on the miniImageNet test dataset. For each figure, we mimic the test procedure by randomly selecting five classes from the test dataset. Then, we compute the features of 200 samples per class and create the visualizations of the features shown in each figure using t-SNE [74], where we use the same 200 samples for different methods in each figure.

From Figure 6, we can see that our feature embedding can well separate the five classes, especially for cuirass, crate, and mixing bowl, as compared to Proto Net and Relation Net. Although it is quite challenging to distinguish the two species of dogs shown in the unseen novel classes, our method can still better separate their features compared with Proto Net and Relation Net. Since the feature embeddings are visualized on the novel classes, the results clearly demonstrate that our method produces better feature embeddings on the novel classes compared to the other two methods. Therefore, our results make it easier for the subsequent K-NN to perform the classification, thus leading to more promising few-shot classification results. Please see more visualized results in the section 2 in the supplementary file.

6. Conclusion

In this work, we revisit the metric learning and propose a simple and effective K-tuplet network for few-shot learning. We present an efficient K-tuplet network to utilize the relationship of training samples to learn the transferable feature embedding that performs well not only on the training samples but also on the novel class samples. Built on top of this generalized feature embedding, we can largely improve the few-shot classification accuracy. Our method is simple yet effective, and outperforms other metric-based few-shot classification algorithms on the public benchmark dataset. More importantly, our method can generalize very well to the novel categories even on other four datasets.

References

[1] C. Zhang, C. Li, J. Cheng, Few-shot visual classification using image pairs with binary transformation, IEEE Transactions on Circuits and Systems for Video Technologydoi:10.1109/TCSVT.2019.2920783.

[2] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: NIPS, 2017, pp. 4077–4087.

[3] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: ICML, 2017, pp. 1126–1135.

[4] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: ICLR, 2017.

[5] D. Wertheimer, B. Hariharan, Few-shot learning with localization in realistic settings, in: CVPR, 2019, pp. 6558–6567.

[6] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, J. Luo, Revisiting local descriptor based image-to-class measure for few-shot learning, in: CVPR, 2019.

[7] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, J.-B. Huang, A closer look at few-shot classification, in: ICLR, 2019.

[8] L. Bertinetto, J. F. Henriques, P. Torr, A. Vedaldi, Meta-learning with differentiable closed-form solvers, in: ICLR, 2019.

[9] M. A. Jamal, G.-J. Qi, M. Shah, Task-agnostic meta-learning for few-shot learning, in: CVPR, 2019.

[10] H. Li, D. Eigen, S. Dodge, M. Zeiler, X. Wang, Finding task-relevant features for few-shot learning by category traversal, in: CVPR, 2019.

[11] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10657–10665.

[12] J. Kim, T. Kim, S. Kim, C. D. Yoo, Edge-labeling graph neural network for few-shot learning, in: CVPR, 2019, pp. 11–20.

[13] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, S. X. Yu, Large-scale long-tailed recognition in an open world, in: CVPR, 2019, pp. 2537–2546.

[14] Y. Lifchitz, Y. Avrithis, S. Picard, A. Bursuc, Dense classification and implanting for few-shot learning, in: CVPR, 2019.

[15] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, Y. Yang, Learning to propagate labels: Transductive propagation network for few-shot learning, in: ICLR, 2018.

[16] Q. Sun, Y. Liu, T.-S. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: CVPR, 2019, pp. 403–412.

[17] Z. Chen, Y. Fuy, Y. Zhang, Y.-G. Jiang, X. Xue, L. Sigal, Multi-level semantic feature augmentation for one-shot learning, IEEE Transactions on Image Processing.

[18] Z. Chen, Y. Fu23, K. Chen, Y.-G. Jiang123, Image block augmentation for one-shot learning, Vol. 6, 2019.

[19] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, M. Hebert, Image deformation meta-networks for one-shot learning, in: CVPR, 2019, pp. 8680–8689.

[20] L. Long, W. Wang, J. Wen, M. Zhang, Q. Lin, B. C. Ooi, Object-level representation learning for few-shot image classification, arXiv:1805.10777.

[21] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, R. Feris, A. Kumar, R. Giryes, A. M. Bronstein, Delta-encoder: an effective sample synthesis method for few-shot object recognition, arXiv:1806.04734.

[22] B. N. Oreshkin, A. Lacoste, P. Rodriguez, Tadam: Task dependent adaptive metric for improved few-shot learning, arXiv:1805.10123.

[23] A. Li, T. Luo, Z. Lu, T. Xiang, L. Wang, Large-scale few-shot learning: Knowledge transfer with class hierarchy, in: CVPR, 2019, pp. 7212–7220.

[24] Y. Liu, Q. Sun, A.-A. Liu, Y. Su, B. Schiele, T.-S. Chua, Lcc: Learning to customize and combine neural networks for few-shot learning, in: CVPR, 2019.

[25] C. Zhang, Q. Huang, Q. Tian, Contextual exemplar classifier-based image representation for classification, IEEE Transactions on Circuits and Systems for Video Technology 27 (8) (2016) 1691–1699.

[26] M. A. Jamal, G.-J. Qi, Task agnostic meta-learning for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11719–11727.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[28] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Artificial Intelligence and Statistics, 2015, pp. 562–570.

[29] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: ICML, 2016, pp. 1842–1850.

[30] T. Munkhdalai, H. Yu, Meta networks, in: ICML, 2017, pp. 2554–2563.

[31] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, in: ICML Deep Learning Workshop, Vol. 2, 2015.

[32] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: NIPS, 2016, pp. 3630–3638.

[33] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, T. M. Hospedales, Learning to compare: Relation network for few-shot learning, in: CVPR, 2018.

[34] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 594–611.

[35] B. Lake, R. Salakhutdinov, J. Gross, J. Tenenbaum, One shot learning of simple visual concepts, in: Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 33, 2011.

[36] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, ICLR.

[37] N. Mishra, M. Rohaninejad, X. Chen, P. Abbeel, A simple neural attentive meta-learner, in: ICLR, 2018.

[38] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375.

[39] S. Qiao, C. Liu, W. Shen, A. L. Yuille, Few-shot image recognition by predicting parameters from activations, in: CVPR, 2018.

[40] A. Bellet, A. Habrard, M. Sebban, A survey on metric learning for feature vectors and structured data, arXiv preprint arXiv:1306.6709.

[41] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: CVPR, 2006.

[42] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: International Workshop on Similarity-Based Pattern Recognition, Springer, 2015, pp. 84–92.

[43] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: NIPS, 2016, pp. 1857–1865.

[44] Y. Liu, J. Lee, M. Park, S. Kim, Y. Yang, Transductive propagation network for few-shot learning, in: ICLR, 2019.

[45] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: CVPR, Vol. 1, 2005.

[46] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, in: CVPR, 2014.

[47] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: CVPR, 2015.

[48] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: CVPR, 2014.

[49] J. Lu, J. Hu, Y.-P. Tan, Discriminative deep metric learning for face and kinship verification, IEEE Transactions on Image Processing 26 (9) (2017) 4269–4282.

[50] J. Hu, J. Lu, Y.-P. Tan, Sharable and individual multi-view metric learning, IEEE transactions on pattern analysis and machine intelligence 40 (9) (2017) 2281–2288.

[51] T. Xiao, S. Li, B. Wang, L. Lin, X. Wang, Joint detection and identification feature learning for person search, in: CVPR, 2017.

[52] Y. Duan, J. Lu, W. Zheng, J. Zhou, Deep adversarial metric learning, IEEE Transactions on Image Processing 29 (2020) 2037–2051.

[53] Z. Wu, A. A. Efros, X. Y. Stella, Improving generalization via scalable neighborhood component analysis, in: ECCV, Springer, 2018, pp. 712–728.

[54] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, R. S. Zemel, Meta-learning for semi-supervised few-shot classification, in: ICLR, 2018.

[55] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, in: ICLR, 2018.

[56] H. Li, W. Dong, X. Mei, C. Ma, F. Huang, B.-G. Hu, Lgm-net: Learning to generate matching networks for few-shot learning, in: ICML, 2019.

[57] A. Mehrotra, A. Dukkipati, Generative adversarial residual pairwise networks for one shot learning, arXiv:1703.08033.

[58] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, R. S. Zemel, Meta-learning for semi-supervised few-shot classification, in: ICLR, 2018.

[59] H. Li, D. Eigen, S. Dodge, M. Zeiler, X. 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.

[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.

[61] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.

[63] T. Munkhdalai, X. Yuan, S. Mehri, A. Trischler, Rapid adaptation with conditionally shifted neurons, in: ICML, 2018.

[64] M. Ye, Y. Guo, Deep triplet ranking networks for one-shot recognition, arXiv:1804.07275.

[65] Y. Wang, X.-M. Wu, Q. Li, J. Gu, W. Xiang, L. Zhang, V. O. Li, Large margin few-shot learning, arXiv:1807.02872.

[66] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, Cosface: Large margin cosine loss for deep face recognition, in: CVPR, 2018, pp. 5265–5274.

[67] B. Recht, R. Roelofs, L. Schmidt, V. Shankar, Do cifar-10 classifiers generalize to cifar-10?, arXiv preprint arXiv:1806.00451.

[68] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Computer vision and Image understanding 106 (1) (2007) 59–70.

[69] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-UCSD Birds 200, Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010).

[70] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization, in: First Workshop on Fine-Grained Visual Categorization, CVPR, Colorado Springs, CO, 2011.

[71] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

[72] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012, pp. 1097–1105.

[73] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: ICLR, 2015.

[74] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.