Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Temperature network for few-shot learning with distribution-aware large-margin metric

Wei Zhu[a], Wenbin Li[b], Haofu Liao[a], Jiebo Luo[a,*]

[a] Department of Computer Science, University of Rochester, Rochester NY 14627, USA
[b] Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China

ARTICLE INFO

Article history:
Received 6 October 2019
Revised 2 November 2020
Accepted 13 December 2020
Available online 6 January 2021

Keywords:
Few-shot learning
Metric learning
Skin lesion classification
Temperature function

ABSTRACT

Few-shot learning learns to classify unseen data with few training samples in hand and has attracted increasing attentions recently. In this paper, we propose a novel *Temperature Network* to tackle few-shot learning tasks motivated by three crucial factors that are seldom considered in the existing literature. First, to encourage compact intra-class distribution, a general improvement for prototype-based methods is proposed to ensure compact intra-class distribution and the effectiveness is theoretically and experimentally validated. Second, the proposed Temperature Network can implicitly generate query-specific prototypes and thus enjoys a more effective distribution-aware metric. Third, to further strengthen the generalization ability of the proposed model, a novel and simple large-margin based method is developed by leveraging the temperature function and we gradually tune the learning temperature to stabilize the training process. Moreover, we note that the commonly used datasets in few-shot learning are actually contrived from large-scale datasets, and thus may not represent a real few-shot problem. We propose a real-life few shot problem, *i.e.*, *Dermnet skin disease*, to comprehensively evaluate the performance of few-shot learning methods. Experiments conducted on conventional datasets as well as the proposed skin disease dataset demonstrate the superiority of the proposed method over other state-of-the-art methods. The source code of our method is available.[1]

1. Introduction

Recently, deep neural networks show superior performance in the fields of machine learning, computer vision, and natural language processing, *etc*. However, conventional deep neural networks are prone to overfitting, and thus generally requires a large number of labelled data [1,2]. However, it is usually difficult or even prohibitive to collect a large and labeled dataset in real life applications (*e.g.*, medical images). The limited dataset size poses great challenges to the generalization ability of the conventional methods [3], and thus initiates the development of few-shot learning techniques [4]. There are mainly three types of few-shot learning methods, including metric-learning based methods (“learning to compare”) [5–9], meta-learning based methods (“learning to learn”) [10–12], and hallucination based methods (“learning to augment”) [13–15]. Among them, metric-learning based methods attract wide interests due to their simplicity and effectiveness. Its basic idea is to learn a metric space that has large inter-class and small intra-class distance.

Despite the great success of the metric-learning methods, existing approaches have several limitations. First, classical prototype-based methods [6,7,16] oversimplify the distance calculation between the query and categories by representing each category with a center point. Then the distance between the query and the category is calculated by the distance between the query and the center point. However, this oversimplified approach cannot guarantee compact intra-class distributions. To address this issue, we make a simple yet effective improvement to punish the scatter distributions. Specifically, we alternatively calculate the distance between the query and the category using the average distance between the query and the category’s support samples. The effectiveness of the improvement is theoretically and empirically validated.

Second, most of the metric learning-based methods only generate a single prototype for each category by treating all support samples of this category equally. Given a query sample, the support samples that are closer to the query are more likely to contain relevant and important information with regarding to the query and thus should contribute more to its classification. This intuition is consistent with many classical methods, such as *k*-Nearest Neigh-

* Corresponding author.
E-mail addresses: zwvews@gmail.com (W. Zhu), liwenbin.nju@gmail.com (W. Li), haofu.liao@rochester.edu (H. Liao), jluo@cs.rochester.edu (J. Luo).

[1] https://github.com/zwvews/TemperatureNetwork.git

https://doi.org/10.1016/j.patcog.2020.107797
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

bor classifier, RBF-Network, RBF SVM, and so on [17]. In terms of metric learning, the idea is even more popular and the local metric usually performs better than the global metric according to the low-dimensional manifold assumption [18–20]. A few methods try to capture the relevant information by an attention-based network. For instance, Matching Net [5] introduces full context embeddings (FCE) and utilizes a LSTM [21] with read-attention to filter related information contained in support samples. Similarly, simple neural attentive learner (SNAIL) [22] proposes a soft-attention strategy [23] to extract relevant information. Ideally, such methods are possible to extract query-specific information while eliminating irrelevant information contained in support samples. However, we note that it is potentially unpractical and ineffective especially in the scenario of few-shot learning. In this paper, we propose a *Temperature Network* by taking advantage of the temperature function to implicitly generate query-specific prototypes. We make best use of the prior knowledge to re-weight the support samples automatically based on their distances to the query. We find this leads to a more effective distribution-aware local metric.

Moreover, due to the fact that the generalization ability is crucial for few-shot learning with extremely scarce training samples available, in this paper, an effective large-margin training strategy is proposed to further enhance the proposed Temperature Network. The basic idea is that, during training, we intentionally make query samples that are not close enough to the positive category harder to be correctly classified. Therefore, the learned metric space is forced to be more discriminative in order to minimize the classification loss. This actually works similarly to classical large-margin methods [24–27], which also have been introduced to few-shot learning recently [28]. However,the proposed method is superior to conventional large-margin regularized few-shot methods [28] in several ways. First, we directly enhance the learned distance metric and do not complicate the loss function with additional large-margin regularization terms. Moreover, our method does not require additional construction of the triplets or pairs which are essential and crucial to conventional large-margin based methods [28–30]. Moreover, instead of setting different temperature for different categories at the very beginning, we gradually tune the temperature to make the training process hard which enables the proposed method to progressively refine the learned metric. The proposed training strategy can also be regarded as a novel type of self-paced learning with respect to the similar motivations, *i.e.*, learning from easy to hard [31,32]. Experimental results also show that it is not proper to directly train with hard metric which may make the network hard to converge, please refer to Table 2 for detail.

Another crucial issue for current few-shot learning methods is that they are all evaluated on contrived datasets (*e.g., mini*ImageNet [33]) and their performance in real-life problems cannot be fairly compared. Therefore, we propose to address a real few-shot learning problem, *i.e.*, Dermnet skin disease classification[2], to comprehensively evaluate the performance of the existing methods. Also, this new real dataset can work as a benchmark dataset in the future. Details of this dataset is provided in Appendix A.

The main contributions of this paper are as follows:

1. A simple and general approach is proposed to enhance the prototype-based few-shot learning methods, which can theoretically lead to compact intra-class distributions.
2. We propose the Temperature Network which can implicitly generate query-specific prototypes. Moreover, in order to best utilize limited training samples, we further propose to train in a "hard mode" to exhaustively mine the large-margin metric.
3. We conduct comprehensive experiments on several public available datasets as well as the proposed Dermnet skin disease dataset to validate the proposed method.

2. Related work

2.1. Few-shot learning

Metric-learning based methods is motivated by the success of deep metric learning. Existing metric learning methods are derived from embedding-based losses, e.g., contrastive loss [24], triplet loss [34], and margin loss [35], etc. Besides the intermediate applications including face verification [36], person re-identification [37–39], and information retrieval [40], metric learning is also widely applied to address the few-shot learning problem.

Generally, metric-learning based few-shot learning methods try to learn a metric embedding space that can transfer the common representations. For example, Matching network [5] adopts cosine similarity to calculate the similarity between queries and support categories. Prototypical network [6] first obtains the center of each category as its prototype and then calculates an Euclidean distance between the prototype and the query sample. Sung et al. propose Relation net [7] to directly learn the embedded metric space. Graph Neural Network is also applied to few-shot learning to learn the similarity between query and support samples [16,41]. Covariance Metric Network (CovaMNet) [42] adopts the covariance matrix to exploit the second-order information. However, the metric space learned by these methods only restrain a large inter-class distance constraint but cannot guarantee compact intra-class distribution. Moreover, most of these methods treat support samples equally and the obtained universal metric may not be suitable for various queries.

Meta-learning based methods are trained to directly learn an optimizer or initialization over a batch of tasks. Model-Agnostic Meta-Learning (MAML) proposed by Finn et al. [10] aims at learning a good model initialization which allows the network to deal with the scenario of limited training samples. Another representative work, LSTM-based meta learnerd [11] tries to learn to emulate stochastic gradient descent algorithm which then can be directly used to optimize new coming tasks. Recently, Rusu et al. [12] propose a latent embedding optimization algorithm to learn a latent representation which can be used to perform gradient descent.

Hallucination based methods learn the rules to augment data according to the auxiliary set. Most of these methods are based on auto-encoders [43] or generative adversarial networks [44]. Clearly, hallucination based methods can be jointly utilized with other few-shot learning methods, and are thus usually not compared with metric-learning based and meta-learning based methods in the literature. [13] proposes a data augmentation GAN to generate new samples for one-shot learning. [15] uses a hallucinator to generate new samples which are then fed into a meta learner combined with the original data [15]. Schwartz et al. [14] utilize an auto-encoder to learn the transformation between samples from the same category to augment the data [14].

Other related few-shot learning methods are also included here for completeness. Semi-supervised few-shot learning, first studied by Ren et al. [8], differs from conventional few-shot learning by utilizing all query samples within each episode. Ren et al. propose to use unlabelled data to facilitate the calculation of the prototypes [8]. Liu et al. [45] recently propose a transductive propagation network by formulating the semi-supervised few-shot learning as a label propagation problem.

[2] https://www.dermnet.com

2.2. Knowledge distillation network

Our method is also inspired by the knowledge distillation network in terms of exerting the superior properties of the exponential function [46]. The essence of distillation network is that, when we increase the temperature, the differences between positive and negative categories can be eliminated to some extent, and the network is thus forced to learn more discriminative representations. [47] conducts omni-supervised learning by distilling knowledge from labelled data. [48] directly distills the dataset to obtain representative samples. [49] proposes to learn features from different compactness levels with different temperatures. However, the proposed method is different from all of these works. First, we tune the temperature in our metric learning layer instead of the softmax layer. Second, during training, we set different temperatures for positive and negative categories respectively, which enables us to obtain a large-margin metric. Moreover, we propose to gradually tune the temperatures to progressively refine the learned metric which can also stabelizes the training process.

2.3. Mining hard samples

Hard sampling is a general strategy to strengthen the model's ability [50]. For example, inspired by the importance sampling, [51] proposes to focus on hard samples to speedup the training process. Harwood et al. propose to construct triplets of close negative and far positive samples with respect to the selected anchors [52]. Zhao et al. [53] propose to generate hard triplets by adversarial learning. In this paper, instead of constructing hard training samples, we assign different temperatures to different categories to tough the training process.

3. Our method

3.1. Problem formulation

For few-shot learning, we are given a support set S, a query set X, and an auxiliary set A, where S contains B different categories and each of them has K training samples, *i.e.,* B-way K-shot. Generally, few-shot learning methods are tested on 5-way 1-shot and 5-way 5-shot scenarios. Clearly, training a classifier solely on the support set hardly achieves reasonable performance with few labeled samples. Therefore, it is crucial to learn and transfer knowledge from the auxiliary set A to support set S, although A shares no common categories either in S or X. At first sight, transfer learning or domain adaption methods may properly handle the task [54,55]. However, as pointed by Snell et al. [6], the essential problem of transfer learning based methods is that the training condition is different from that of the testing, which will degrade the performance. Snell et al. [6] proposes a novel episode training mechanism, which is validated to be effective by recent papers. In episode training, we generate episodes by drawing some samples from the auxiliary set A, with some of them regarded as support samples S_{train}, others as query samples X_{train}. At each iteration, we train the model with the constructed episodes.

3.2. Improved prototypical network

The Prototypical Network is a well-known few-shot learning method. Given a query sample $x \in R^d$ and a certain category $C = \{c_1, c_2, \ldots, c_\ell\}, c_i \in R^d$, where ℓ is the number of support samples of the category, a Prototypical Network calculates the distance D_1 between query x and category C as

$$D_1(x, C) = M_p(x, \frac{1}{\ell}\sum_{i=0}^{\ell} c_i) \tag{1}$$

where $M_p(\cdot,\cdot)$ denotes the Minkowski distance with order $p > 1$. Clearly, the Prototypical Network, i.e., Eq. (1), cannot guarantee compact intra-class distribution. To alleviate the problem, we propose a new formulation as follows

$$D_2(x, C) = \frac{1}{\ell}\sum_{i=0}^{\ell} M_p(x, c_i). \tag{2}$$

We call this Improved Prototypical Network which has the following property.

Lemma 1. *For a Minkowski distance with order* $p > 1$, $D_1(x, C) \leq D_2(x, C)$ *and the equality holds if and only if* $c_i = \frac{1}{\ell}\sum_{i=0}^{\ell} c_i$ *for any* $i \in \{1, 2, \ldots, \ell\}$.

Detailed proof is provided in Appendix B, and is based on the fact that the Minkowski distance, except the Manhattan distance where $p = 1$, can be derived by a strictly convex ℓ_p vector norm. For the Manhattan distance, $c_i = \frac{1}{\ell}\sum_{i=0}^{\ell} c_i$ is only a sufficient but not necessary condition for the equality since the ℓ_1 norm is not strictly convex. To give an intuitive understanding of Lemma 1, taking the original Prototypical Network as an example where the Minkowski distance is an Euclidean distance M_2, we have

$$D_2(x, C) - D_1(x, C) = var(c_i), \tag{3}$$

where the variance of category C is

$$var(c_i) = \frac{1}{\ell}\sum_{i=1}^{\ell} \|c_i - E(c_i)\|_2^2 \geq 0 \tag{4}$$

Detailed derivation is shown in Appendix C. The equality, i.e., $D_1(x, C) = D_2(x, C)$, holds if and only if $var(c_i) = 0$. Therefore, given two different distributions with the same center, $D_1(s, C)$ will consider them as identical while $D_2(s, C)$ will favor the compact one whose $var(c_i)$ is smaller. In other words, $D_2(s, C)$ punishes large intra-class scatter and thus leads to a compact distribution.

3.3. Temperature network for few-shot learning

The Temperature Network is proposed to alleviate several limitations of existing metric learning-based methods. First of all, our method can generate query-specific prototypes. Moreover, a novel large margin-based method is proposed to train the model on "harder" scenarios for better generalization ability. An overview of the proposed Temperature Network is shown in Fig. 1. An intuitive illustration to show the superiority of the proposed method is present in Fig. 2.

3.3.1. Temperature network with query-specific prototypes

For a B-way K-shot task, we are given a query sample and a support set. The support set consists of B different categories and each category contains K samples, where K usually ranges from 1 to 5. The feature map extracted from a convolutional neural network is $W \times H \times d$, where W is the width, H is the height, and d is the number of channels. We follow Li et al. [42] to extract local descriptors to perform few-shot learning. The Cth category is represented by $C = \{c_1, c_2, \ldots, c_\ell\}$, where category C is among the total B categories, $c_i \in R^d$ is the ith local descriptor and $\ell = K \times H \times W$. Then, given a query sample $X \in R^{W \times H \times d} = \{x_1, x_2, \ldots, x_{W \times H}\}$ after feature extraction, we first calculate the similarity between local descriptor x_i and c_j as follows

$$s(x_i, c_j) = f(d(x_i, c_j)) = \exp\left(-\frac{d(x_i, c_j)}{T}\right), \tag{5}$$

where T denotes the temperature which is fixed as 10 in our experiments otherwise stated, and $d(x_i, c_j)$ denotes the distance measurement between x_i and c_j, *e.g.,* order p Minkowski distance $M_p(\cdot,\cdot)$. It is worth noting that $f(d(x_i, c_j))$ has several good

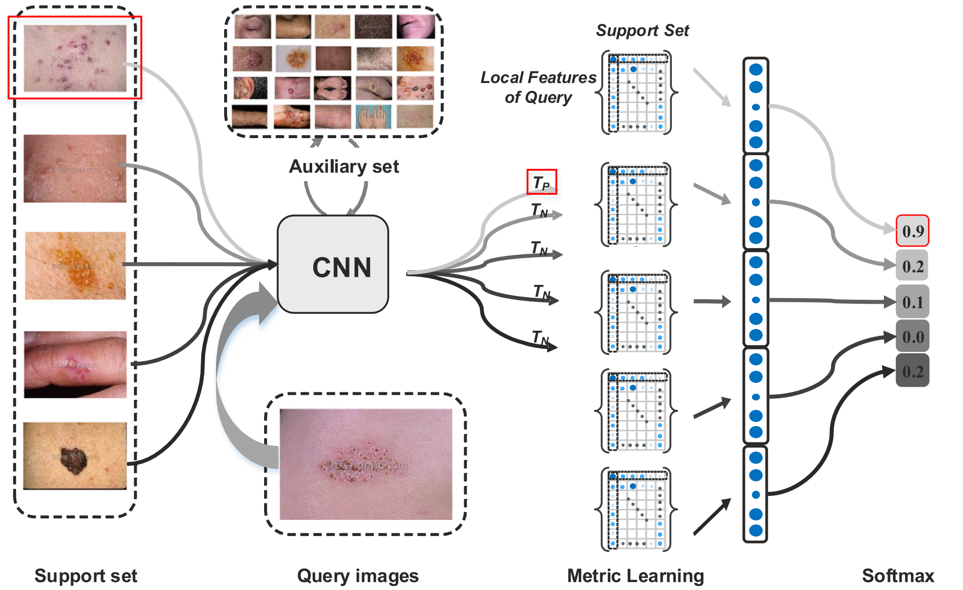


Fig. 1. An overview of Temperature Network. We take the proposed Dermnet skin disease dataset as an example. Different temperature is assigned for positive and negative categories during training, and the same temperature is set during testing.

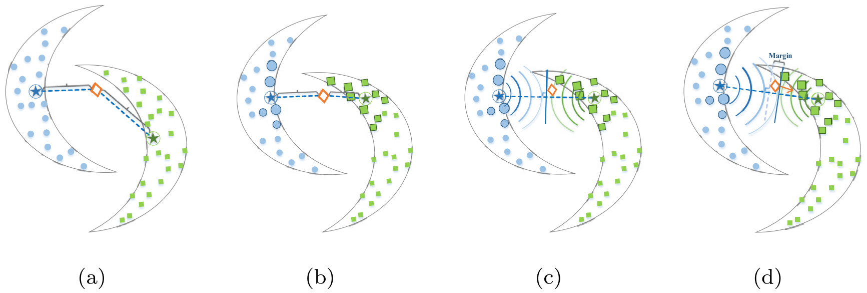


Fig. 2. Illustration of the similarity measurements of different methods. We denote the orange diamonds as the queries, stars as the generated prototypes and others as support samples. (a) Prototypes generated by Prototypical Network are fixed for different queries; (b) Prototypes implicitly generated by the Temperature Network are query-specific; (c) Temperature Network with the same temperature for all the categories may not be able to make a max-margin metric; (d) Large-margin metric could be obtained with class-specific temperatures during training for better test (generalization) performance.

properties: (1) $0 < f(d(x_i, c_j)) \leq 1$; (2) with finite temperature T, $f(d(x_i, c_j)) = 1$ if and only if $d(x_i, c_j) = 0$; (3) $f(d(x_i, c_j))$ is monotonically decreasing; (4) the gradient norm of $f(d(x_i, c_j))$ is monotonically decreasing.

Then, the similarity between x_i and the Cth category can be calculated by averaging the point-wise similarity as

$$s_{mean}(x_i, C) = \frac{1}{\ell} \sum_{j=1}^{\ell} s(x_i, c_j). \tag{6}$$

The Temperature function, i.e., the Gaussian kernel function $f(d(x_i, c_j)) = \exp(-\frac{d(x_i, c_j)}{T})$, is used to simultaneously calculate the similarity and re-weight the support samples based on their distances to the query sample to generate query-specific prototypes. To see how Temperature function works, according to Fig. 3, since the gradient norm of $f(d(x_i, c_j))$ decreases monotonically, to make the similarity $s(x_i, C)$ large, our method will put more weights on c_j's that are close to x_i. That is, the support samples are automatically re-weighted according to their distance to x_i which eventually lead to query-specific prototypes. We note that temperature T can control the extent of the specific and locality of the generated prototypes. To make it clear, considering following two extreme cases, if T approaches infinity, $s(x_i, c_j)$ will approach 1 for any x_i and c_j, and thus $s_{mean}(x_i, C)$ will be influenced almost equally by all x_i, c_j pairs. On the other hand, if T approaches zero, $s_{mean}(x_i, C)$ is only influenced by c_j corresponding to the largest $s(x_i, c_j)$. Therefore, high/low temperature takes more/less support samples into consideration. We will further utilize this property in Section 3.3.2 to boost out model.

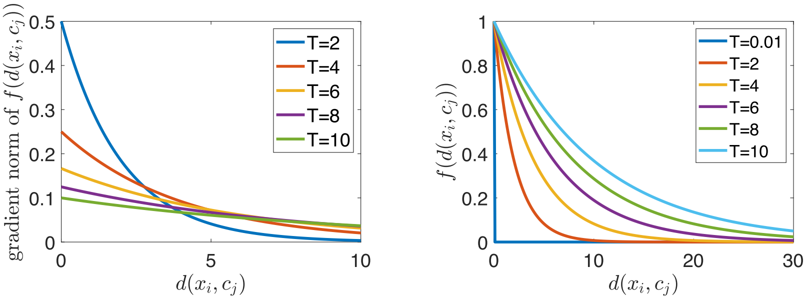
Note that, unlike the Prototypical Network [6], our method does not explicitly generate prototypes. In contrast, the mean similarity $s(x_i, C)$ calculated by Eq. (6) can be seen as the distance between x and a virtual query-specific prototype as shown in Fig. 2(b). Moreover, the formulation of Eq. (6) provides convenience for other statistic calculation. For example, the standard deviation can be calculated as

$$s_{std}(x_i, C) = \left(\frac{1}{\ell} \sum_{j=1}^{\ell} (s(x_i, c_j) - s_{mean}(x_i, C))^2\right)^{\frac{1}{2}}. \tag{7}$$

Then, we can jointly utilize the mean and standard deviation to calculate the similarity metric between x_i and C as

$$s(x_i, C) = s_{mean}(x_i, C) * (s_{std}(x_i, C))^{\rho}, \tag{8}$$

where $\rho > 0$ is a hyper-parameter. Basically, according to Eq. (8), the similarity $s(x_i, C)$ is large when the average similarity $s_{mean}(x_i, C)$ is large and the standard deviation $s_{std}(x_i, C)$ is large. A large standard deviation means that some local descriptors x_i are relatively closer to c_j than others, and in other words, some local descriptors x_i would be extremely close to c_j with a large average similarity. It is worth mentioning that this does not conflict with Lemma 1 which forces the distribution of C to be compact, since $s_{std}(x_i, C)$ exerts influences on the similarity between x_i and c_j. Although superior performance could be obtained by tuning the



(a) Gradient of $f(d(x_i,c_j))$	(b) $f(d(x_i,c_j))$

Fig. 3. Our method makes advantage of the temperature function in terms of function value and gradient norm. To generate query-specific prototypes, the support samples are automatically re-weighted based on their distance to query by exerting the monotonically decreasing property of the gradient norm of $f(d(x_i, c_j))$ as shown in (a). For the category-specific temperature setting, since $T_P \le T_N$, to have high similarity for positive category, based on (b), positive support samples are required to be much closer to the query.

value of hyper-parameter ρ, we fix $\rho = 1$ in our experiments unless otherwise stated.

3.3.2. Class-specific temperature

As mentioned earlier, the scarcity of training samples in few-shot learning poses huge challenges to the generalization ability of learning models. It turns out that a simple modification on Eq. (5) can greatly enhance the performance. During training, we manually specify different temperatures for the positive and negative categories respectively. For detail, given a query sample x belonging to the C^xth category, $s(x_i, c_j)$ is calculated by

$$s(x_i,c_j)=\begin{cases}\exp\left(-\frac{d(x_i,c_j)}{T_P}\right), & C = C^x\\ \exp\left(-\frac{d(x_i,c_j)}{T_N}\right), & C \neq C^x\,,\end{cases} \tag{9}$$

where T_P is the temperature for the positive category, i.e., the ground-truth category, and T_N is that for other categories, and $T_P \le T_N$. According to Eq. (9), given a positive category C^P and a negative one C^N, even though every $d(x_i, c_j^P)$ is just slightly smaller than $d(x_i, c_j^N)$, we still have $s(x_i, C^P) \le s(x_i, C^N)$ considering $T_P \le T_N$ (see Fig. 3(b)), and thus will have large losses and potentially lead to misclassification of the query sample. That is, during training, our model will push the query much closer to the positive category, i.e., making $d(x_i, c_j^P)$ significantly smaller than $d(x_i, c_j^N)$. This consequently leads to a large-margin metric and will enhance the generalization performance. As for the implementation, we first set same temperature for both positive and negative categories, and then gradually decrease and increase temperature respectively to enable our method to progressively refine the learned metric. It is also experimentally validated that this can stabilize the training process compared with setting different temperatures at the very beginning, see Table 2 for detailed results. In practice, the temperature is initialized as 10 for both T_P and T_N and will be gradually tuned every $\frac{5n}{3}$ episodes.

Once $s(X, C) = \{s(x_1, C), s(x_2, C), \ldots, s(x_\ell, C)\}$ is obtained, the overall similarity between the query sample $X = \{x_1, x_2, \ldots, x_{W\times H}\}$ and the Cth category is obtained by a weighted sum as $w^T s(X, C)$. Next, a softmax layer with cross-entropy loss is used to perform the final classification. We summarize the proposed method in Algorithm 1.

3.4. Network architecture

The proposed Temperature Network contains two modules, including one CNN feature extraction module and one temperature metric learning module. For fairness, the adopted feature extraction module contains 4 convolution blocks with each containing the sequence of a convolutional layer with 64 filters of size 3 × 3, a batch normalization and a Leaky ReLU layer. The first two blocks contain an extra 2 × 2 max-pooling layer, respectively. Similar network architectures are commonly adopted by most of few-shot learning methods as a benchmark to test the effectiveness of the following metric learning module. For the metric learning module, we choose $d(x, y)$ as Euclidean distance, and when calculating the standard deviation, we add a small constant (i.e., 1×10^{-7}) to prevent the gradients from NaN.

Algorithm 1 Temperature network for few shot learning.

```
Input: Initialization Temperature $T_P = T_N = 10$, temperature step
    size $\delta_P = 0.5$ and $\delta_N = 1.5$, $\rho = 1$;
for each episode do
    STEP 1: Feature Embedding
    Extract features for query $X \in \{x_1, x_2, \ldots, x_{W\times H}\}$ and support
    samples of each category;
    STEP 2: Metric Learning
    for each category $C$ do
        for each local descriptor $x_i$ of query do
            Calculate the similarity $s(x_i, c_j)$ between local descrip-
            tor $x_i$ and $c_j$ with $T_P$ and $T_N$ by Eq. (9), where $c_j$ is the
            $j$th local descriptor of support category;
            Calculate $s_{mean}(x_i, C)$ by Eq. (6);
            Calculate $s_{std}(x_i, C)$ by Eq. (7);
            The final similarity $s(x_i, C)$ between $x_i$ and category $C$
            is obtained by Eq. (8);
        end for
        Calculate the final similarity between $X$ and $C$ by
        a weighted sum of local descriptor-based similarity
        $\{s(x_1, C), s(x_2, C), \ldots, s(x_\ell, C)\}$;
    end for
    STEP 3: Classification
    Conduct Classification via softmax loss and backpropagation
    to update all parameters end-to-end;
    STEP 4: Temperature Tuning
    Tune Temperature every $r$ episode as
    $T_P = T_P * \delta_P$
    $T_N = T_N * \delta_N$
end for
```

4. Experiments

We validate the effectiveness of the proposed method on several benchmark datasets including the popular few-shot classification dataset *mini*ImageNet, two fine-grained datasets, *i.e.*, Stanford

Dogs and Stanford Cars, and the proposed Dermnet skin disease dataset.

4.1. Datasets and settings

We briefly describe the used datasets as follows:

1. ***mini*Imagenet** [33] contains 100 categories that are selected from ImageNet [56], and each category consists of 600 images [5]. We follow the popular splits proposed by Ravi and Larochelle [11], i.e., 64 categories for training, 16 for validation, and 20 for testing. For fine-grained datasets,
2. **Stanford Dogs** dataset [57] consists of 20580 images and 120 categories and 70, 20, and 30 categories are used for training, validation, and testing.
3. **Stanford Cars** dataset [58] has 16185 images from 196 categories and we follow [42] to make 130, 17, 49 categories for training, validation, and testing.
4. **Dermnet skin disease** dataset contains 20230 images and 334 categories. We manually split 186 categories for training, 74 for validation and testing respectively. For detail description, please refer to Appendix A.

Experimental settings are same and fixed for all dataset otherwise stated. We conduct 5-way 1-shot and 5-way 5-shot tasks on all datasets. 300,000 episodes are constructed to train our model and each episode contains 5 categories and each category has additional 15 query samples. We adopt Adam algorithm with the initial learning rate of 0.001 which will be cut by 0.1 for every $\frac{5n}{3}$ episodes. The temperature T_P and T_N is initialized as 10 for all experiments otherwise stated and is multiplied by $\delta_P = 0.5$ and $\delta_N = 1.5$ for positive and negative categories every $\frac{5n}{3}$ episode respectively, where δ_P and δ_N are the positive and negative temperature step size respectively. Moreover, ρ is default set as $\rho = 1$. For testing, the temperature is set as T_p. We report the mean classification accuracy by constructing 600 episodes from the testing set.

4.2. Compared methods

To fully demonstrate the superior performance of the proposed method, ten state-of-the-art few-shot learning methods in addition to K-NN are bench-marked. These methods are briefly shown as follows:

1. *Matching Net FCE* [5] utilizes cosine similarity-based metric to perform few-shot learning. Furthermore, Full Context Embeddings(FCE) is introduced to filter and integrate support information which leads to better performance.
2. Meta-Learner LSTM [11] is proposed to directly learn the optimization algorithm from auxiliary set which then can be directly used to train meta-tasks.
3. Model-Agnostic Meta-Learning (MAML) [10] aims to learn a good initialization for meta-tasks. Few training iterations need to be conducted on the meta-tasks with proper initialization.
4. Prototypical Net (PN) [6] proposes to directly learn prototypes for each category which intuitively lead to better generalization ability.
5. Graph Neural Network [16] is proposed recently and is validated to be effective for various node classification tasks. The few-shot learning is explained as a label propagation task and thus can be directly tackled by GNN.
6. Simple Neural Attentive Learner (SNAIL) [22] utilizes temporal convolution and soft attention which works similarly as the FCE used in Matching Net. The temporal convolution integrates information of support data which the useful information is then selected by soft attention.
7. Large-margin Prototypical Net (L-PN) [28] equips large-margin regularization terms to Prototypical Net. The method needs additional triplet construction similar as triplet loss and we select cosine distance for its better performance according to Wang et al. [28].
8. Large-margin Graph Neural Network (L-GNN) [28] adopts Normface Loss as large-margin terms to enhance GNN. The method also requires additional triplet construction and introduces several hyper-parameters needed to be tuned.
9. Relation Net [7] contains a neural network-based relation module. That is, the metric is obtained through a learnable network.
10. Covariance Metric Net (CovaMNet) [42] takes the second order information into consideration and obtains the similarity through Covariance Metric.

We do not include PN results with the high way trick reported in the original work for fairness and its results are courtesy of [42] (Stanford Dogs and Stanford Cars) and [59] (*mini*ImageNet), respectively. The first baseline, K-NN, is conducted on the features extracted by a well-trained CNN network. For the methods proposed in this paper, the improved PN denotes the improved Prototypical Network described in Section 3 and the configuration of Temperature Network is demonstrated previously. Temp Net w/ Temp=10 denotes Temperature Network with fixed temperature for both T_N and T_P. All of these methods are trained with similar 4 convolutional modules without residual connection and dropout to provide a fair comparison. For detail network architecture, please refer to supplementary materials. We here do not adopt advanced backbones, *e.g.,* ResNet and Wide ResNet, since all compared methods are based on 4 convolutional backbone model. Moreover the various deep backbones used by existing literatures also bring difficulties for fair comparison.

4.3. Classification results on commonly-used datasets

The results of *mini*ImageNet, Stanford Dogs, and Stanford Cars, are shown in Table 1. We conclude following interesting points. First, the improved PN does boost the performance of conventional PN in most cases and the amelioration proposed in Section 3.2 is thus potentially able to enhance other PN-based methods. Second, the proposed Temperature Network achieves superior performance on almost all datasets compared with the existing methods. Specifically, the Temperature Network achieves 0.25%, 1.22%, 0.79% improvements over the best existing methods in terms of 5-way 1-shot learning for these three datasets, respectively. For 5-way 5-shot, the Temperature Network obtains 0.33%, 2.51%, 0.24% improvements over the state-of-the-arts methods. The better performance of the proposed methods should be contributed to following points. First, the proposed Temperature Net implicitly generate query-specific prototypes which naturally lead to local and distribution-aware metric. Second, by class-specific temperatures, our method is able to learn a large-margin metric and thus obtains better generalization ability. Moreover, as mentioned, the initialized temperature T for our Temperature Net is meaningful and can be tuned easily for better performance. For example, higher temperature, i.e., large T, allow us to take more support samples into consideration which is thus suitable for high shot tasks. Similarly, low temperature allows us to focus on the core samples and is thus proper for low-shot tasks. Experimental results show that Temperature Net can achieve 52.39% for 5-way 1-shot on *mini*ImageNet.

4.4. Ablation studies

In this section, we conduct ablation studies to show the influence of different parts of Temperature Net for final performance.

Table 1
5-way accuracy of all methods on three datasets. ‡ denotes Large-margin PN with cosine distance for its better performance. † denotes Large-margin GNN using Normface loss with $s = 10, \lambda = 1, m = 0.5$. § denotes that results of this dataset are retrieved from Li et al. [42]. ⋆ denotes results are retrieved from Chen et al. [59]. £ denotes the experiments are exactly same as PN which we omit here. Other results are retrieved from their original work.

Method	Stanford Dogs§		Stanford Cars§		miniImageNet	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline k-NN	26.14	43.14	23.50	34.45	27.23	49.29
Meta-Learner [11]	-	-	-	-	43.44	60.60
MAML [10]	-	-	-	-	48.70	63.11
SNAIL [22]	-	-	-	-	45.10	55.20
Matching Net [5]	35.80	47.50	34.80	44.70	43.56	55.31
L-PN(Cosine)‡ [28]	-	-	-	-	50.10	66.94
GNN [16]	46.98	62.27	55.85	71.25	49.02	63.50
L-GNN † [28]	-	-	-	-	51.60	67.25
Relation Net † [7]	-	-	-	-	50.44	65.32
CovaMNet [42]	49.10	63.04	56.65	71.33	51.17	67.65
PN [6]	37.59	48.19	40.90	52.93	44.42⋆	64.24⋆
Improved PN (ours)	-£	52.53	-£	58.73	-£	63.20
Temp Net with fixing Temp 10 (ours)	48.82	61.78	53.37	69.24	51.51	66.52
Temperature Net (ours)	**49.35**	**63.37**	**57.87**	**73.84**	**52.39**	**67.89**

Table 2
Results of different variants of the proposed Temperature Net on *mini* ImageNet for 5-way 1-shot learning.

Method	Temp 10 w/o std	Temp 5/15 w/o std	Temp 10 w/ std
ACC (%)	51.22	50.77	51.51
Method	Temp 5/15 w/ std	Temp Net	
ACC (%)	Not converge	**52.39**	

Table 3
Results with different initial temperature T_P and T_N and temperature step δ_P and δ_N. We set the initial $T_P = T_N$ as shown in the first row and vary δ_P and δ_N from the pairs shown in the first column.

(δ_P, δ_N) (T_P, T_N)	(1,1)	(5,5)	(10,10)	(15,15)	(20,20)
(1,1)	51.15	51.32	51.51	51.86	51.84
(0.8,1.2)	**51.64**	51.59	52.17	51.98	**52.30**
(0.5,1.5)	51.60	51.61	52.39	**52.17**	51.98
(0.5,2)	50.57	**52.40**	**52.57**	51.99	51.87
(0.2,5)	50.76	51.25	51.64	52.01	52.02

Table 4
Results with different ρ for similarity calculation in Eq. (8).

ρ	0.1	0.2	0.5	1	2	5	10
ACC(%)	51.72	51.87	**52.48**	52.39	50.33	49.15	38.10

All experiments are conducted on *mini*ImageNet for 5-way 1-shot case. We first briefly study the effectiveness of the different parts of TempNet in Table 2 and give more detail experiments on the influence of the hyper-parameters in Tables 3 and 4.

We separately test the performance of different variants of our method, including fixed temperature $T = 10$ without the standard deviation (Temp 10 w/o std), fixed temperature $T = 10$ with the standard deviation (Temp 10 w/ std), fixed category-specific temperature ($T_N = 15$ and $T_P = 5$) without standard deviation (Temp 5/15 w/o std), fixed category-specific temperature ($T_N = 15$ and $T_P = 5$) with standard deviation (Temp 5/15 w/ std), and the Temperature Net (Temp Net). The results are shown in Table 2 and we conclude following points. First, the naive Temperature Net with fixed temperature still achieves state-of-the-art performance; second, it is improper to directly set different temperatures at the beginning as the network is hard to converge; third, the inclusion of the standard deviation can improve the performance of our model; last, gradually increasing the difficulty of training strengthens the generalization ability of our model and stabilize the training process.

To investigate the performance of different settings of the class-specific temperature, we vary the initial temperature $T_P = T_N$ from $\{1, 5, 10, 15, 20\}$ and the temperature step pair (δ_P, δ_N) from $\{(1, 1); (0.8, 1.2); (0.5, 1.5); (0.5, 2); (0.2, 5)\}$. The results also shown in Table 3. According to the results, note that $(\delta_P = \delta_N = 1)$ means fixed temperature, better performance could almost always be obtained when using the proposed class-specific temperature training strategy by gradually changing the temperature. This should be attributed to the large margin metric induced by our method. Moreover, it is more desirable to moderately change the temperature, and dramatic changing of the temperature may hurt the performance in practice. We thus set the default temperature step pairs as $(\delta_P, \delta_N) = (0.5, 1.5)$. We'd like to emphasize that this strategy could be directly applied for other tasks to boost their performance. At last, it is desirable to have a large initial temperature by comparing different choices of initial temperature and we set the default initial temperature as $T_P = T_N = 10$ for all datasets.

We also investigate the influence of ρ in Eq. (8) and the results are shown in Table 4. According to the results, excessively large ρ will lead to significant performance degradation, and better performance could be obtained with $\rho \le 1$. It is thus reasonable to set $\rho = 1$ for real life application. It is thus reasonable to set $\rho = 1$ in practical applications.

4.5. Skin disease classification

The skin disease dataset is collected from Dermnet atlas website and contains 20230 images and 334 categories in total. The experimental settings are exactly same as *mini*ImageNet except we change the number of query samples for each category from 15 to 5, since the smallest category contain only 10 iamges. Please refer to supplementary materials for detail description of the dataset.

The resutls are shown in Table 5. PN(E)denotes Prototype Network with Euclidean distance and PN(C) denotes Prototype Network with cosine similarity. We also implement large-margin prototypical network as $L - PN(C)$ and $L - PN(E)$ for Euclidean distance and Cosine similarity respectively. We can conclude that the proposed method outperforms other state-of-the-arts methods a lot in the real-life scenario. For detail, the Temperature Net achieves 4.57% improvements for 5-way 1-shot compared with Relation Net, and obtains 3.32% improvements for 5-way 5-shot com-

Table 5
Classification results on skin disease datasets (ACC %).

Methods	Baseline	Matching Net	PN(E)	PN(C)	L-PN(E)
5-way 1-shot	25.56	44.50	48.57	48.62	48.92
5-way 5-shot	29.16	60.03	66.80	64.20	67.20
Methods	L-PN(C)	GNN	Relation Net	SNAIL	Temp Net
5-way 1-shot	49.27	48.61	48.89	48.25	**53.84**
5-way 5-shot	64.90	68.10	62.37	67.89	**71.42**

pared with the second best model GNN. It is interesting to note that Relation Net does not outperform Prototypical Network in this dataset. The reason may be that the Relation Net benefits from large number of query samples in conventional settings (15 images per categories) by using batch normalization for relation metric module [7,60]. However, there are only 5 query samples available per category for the proposed Dermnet skin disease dataset which may thus essentially harm the learning process for Relation Net. By contrast, the proposed methods is robust to the number of queries and always achieves superior performance.

5. Conclusions and future work

In this paper, we address several limitations for existing few-shot learning methods. First, we propose a general improvement for the popular prototype-based methods which can theoretically lead to compact intra-class distribution. We then propose Temperature Net for few-shot learning. Temperature Net can implicitly generate query-specific prototypes and thus results in local and distribution-aware metric. To further strengthen the generalization ability of the learned metric, we set different temperature for different categories to penalize query samples that are not close enough to their belonging categories. Unlike conventional large-margin metric learning, our method introduces no additional regularization term and also does not need extra triplet/pair constructions. Experiments on benchmark datasets including the proposed skin disease dataset validate the superiority of the method. The ideas adopted by this paper potentially benefit other tasks. For example, besides the general improvements for prototype-based methods, when performing hard sample mining, further improvements are likely obtained by training the network with increasing difficulties. Also, the query-specific metric may also be beneficial to other metric learning-based applications especially for retrieval-based tasks. We will continue our work on these topics in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by NSF award IIS-1722847 and the Morris K. Udall Center of Excellence in Parkinson's Disease Research by NIH.

Appendix A. Skin disease classification

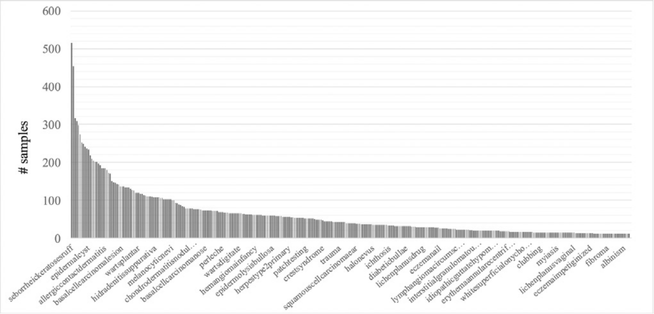


Fig. A1. The category distribution of Dermnet datasets.

For the skin dataset, we collect the dermatology photos from Dermnet atlas website. For detail, we obtain 20230 photos in total which belong to 334 different categories. The category distribution is highly imbalanced. The largest category "seborrheic keratoses ruff" contains 516 photos and the smallest one only has 10 samples. Please refer to Fig. A.4 for detail. To perform few-shot learning, we discard categories with less than 10 samples which are necessary to the 5-way 5-shot setting. The data are manually split into 186 categories for training, 74 for validation and another 74 for testing respectively. Moreover, to better simulate the scenario of few-shot learning, we deliberately make the categories with more than 120 samples (38 categories in total) as training. Please refer to Fig. 1 or the Dermnet Website for sample images.

Appendix B. Proof for Lemma 1

Lemma 1. *For the Minkowski distance, except the Manhattan distance,* $D_1(x,C) \le D_2(x,C)$ *with equality holds if and only if* $c = \frac{1}{\ell}\sum_{i=0}^{\ell} c_i$ *for any* $i \in \{1,2,\ldots,\ell\}$.

Proof. The main idea for the proof is that Minkowski distance can be induced by ℓ_p-norm. Given two vectors x and y, we denote $\|x-y\|_p$ as ℓ_p-norm of vector $x-y$ and is also the Minkowski distance between x and y with the corresponding p be definition, where $p \ge 1$. If $p < 1$, the so-called ℓ_p-norm and p Minkowski distance are no longer norm and distance metric respectively due to the violation of triangle inequality.

Then, according to above definition, we have

$$\begin{aligned} D_1(x,C) - D_2(x,C) &= \|x - \frac{1}{\ell}\sum_{i=0}^{\ell} c_i\|_p - \frac{1}{\ell}\sum_{i=0}^{\ell}\|x-c_i\|_p \\ &= \|\frac{1}{\ell}\sum_{i=0}^{\ell}(x-c_i)\|_p - \frac{1}{\ell}\sum_{i=0}^{\ell}\|x-c_i\|_p \\ &\le \frac{1}{\ell}\sum_{i=0}^{\ell}\|x-c_i\|_p - \frac{1}{\ell}\sum_{i=0}^{\ell}\|x-c_i\|_p \\ &= 0 \end{aligned} \tag{B.1}$$

Here we used the fact that ℓ_p-norm is convex and thus satisfies Jensen's inequality with $p \ge 1$ as shown in Fig B.5. Moreover, if $p > 1$, ℓ_p-norm is strictly convex and and the equality can be fulfilled if and only if $c_i = \frac{1}{\ell}\sum_{i=0}^{\ell} c_i$ for any $i \in \{1,2,\ldots,\ell\}$. If we choose $p = 1$, i.e., using Manhattan distance, the equality clearly always holds. □

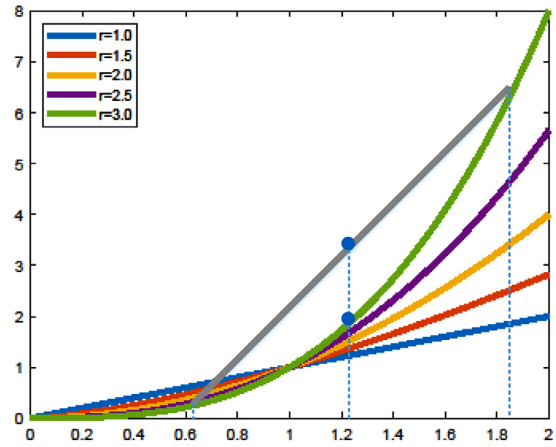


Fig. B1. Illustration of ℓ_p-norm.

Appendix C. Derivation for Eq. (3)

With the square Euclidean distance adopted by original Prototypical Network, the distance $D_1(x, C)$ defined by the Prototypical Network is obtained by the distance between query and the center point as

$$\begin{aligned} D_1(x, C) &= x^T x + \frac{1}{\ell^2}(\sum_{i=0}^{\ell} c_i)^T(\sum_{i=0}^{\ell} c_i) - \frac{2}{\ell} x^T \sum_{i=0}^{\ell} c_i \\ &= x^T x + (E(c_i))^T E(c_i) - 2x^T E(c_i), \end{aligned} \tag{C.1}$$

where $E(c_i) = \frac{1}{\ell}\sum_{i=0}^{\ell} c_i$ is the center of category C. Similarly, for the proposed metric, *i.e.*, Eq. (2), we calculate the average distance between query and support samples as

$$\begin{aligned} D_2(x, C) &= x^T x + \frac{1}{\ell}(\sum_{i=0}^{\ell} c_i^T c_i) - \frac{2}{\ell} x^T \sum_{i=0}^{\ell} c_i \\ &= x^T x + E(c_i^T c_i) - 2x^T E(c_i). \end{aligned} \tag{C.2}$$

We then have

$$\begin{aligned} D_2(x, C) - D_1(x, C) &= E(c_i^T c_i) - (E(c_i))^T E(c_i) \\ &= var(c_i), \end{aligned} \tag{C.3}$$

where $var(c_i) = \frac{1}{\ell}\sum_{i=1}^{\ell} \|c_i - E(c_i)\|_2^2 \geq 0$ is the variance of category C.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.
[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[4] B.M. Lake, R. Salakhutdinov, J. Gross, J.B. Tenenbaum, One shot learning of simple visual concepts, in: Proceedings of the 33th Annual Meeting of the Cognitive Science Society, 2011.
[5] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3630–3638.
[6] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4080–4090.
[7] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H.S. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 1199–1208.
[8] 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, arXiv:1803.00676 (2018).
[9] 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7260–7268.
[10] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, arXiv:1703.03400 (2017).
[11] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning(2016).
[12] A.A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, arXiv:1807.05960 (2018).
[13] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, arXiv:1711.04340 (2017).
[14] 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 (2018).
[15] Y.-X. Wang, R. Girshick, M. Hebert, B. Hariharan, Low-shot learning from imaginary data, arXiv:1801.05401 (2018).
[16] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, CoRR (2017) abs/1711.04043.
[17] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[18] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[19] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput 15 (6) (2003) 1373–1396.
[20] Y. Peng, S. Wang, B.-L. Lu, Marginalized denoising autoencoder via graph regularization for domain adaptation, in: International Conference on Neural Information Processing, Springer, 2013, pp. 156–163.
[21] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[22] N. Mishra, M. Rohaninejad, X. Chen, P. Abbeel, A simple neural attentive meta-learner, arXiv:1707.03141 (2017).
[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[24] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.
[25] F. Wang, X. Xiang, J. Cheng, A.L. Yuille, NormFace: L2 hypersphere embedding for face verification, in: Proceedings of the 25th ACM International Conference on Multimedia, ACM, 2017, pp. 1041–1049.
[26] 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5265–5274.
[27] J. Deng, J. Guo, N. Xue, S. Zafeiriou, ArcFace: additive angular margin loss for deep face recognition, arXiv:1801.07698 (2018).
[28] Y. Wang, X.-M. Wu, Q. Li, J. Gu, W. Xiang, L. Zhang, V.O.K. Li, Large margin few-shot learning, arXiv:1807.02872 (2018).
[29] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking, J. Mach. Learn. Res. 11 (Mar) (2010) 1109–1135.
[30] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, ICML Deep Learning Workshop, vol. 2, 2015.
[31] M.P. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, in: Advances in Neural Information Processing Systems, 2010, pp. 1189–1197.
[32] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, A. Hauptmann, Self-paced learning with diversity, in: Advances in Neural Information Processing Systems, 2014, pp. 2078–2086.
[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.
[34] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2) (2009).
[35] C.-Y. Wu, R. Manmatha, A.J. Smola, P. Krahenbuhl, Sampling matters in deep embedding learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2840–2848.
[36] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[37] C. Zhao, X. Wang, W. Zuo, F. Shen, L. Shao, D. Miao, Similarity learning with joint transfer constraints for person re-identification, Pattern Recognit. 97 (2020) 107014.
[38] C. Zhao, K. Chen, D. Zang, Z. Zhang, W. Zuo, D. Miao, Uncertainty-optimized deep learning model for small-scale person re-identification, Sci. China Inf. Sci. 62 (12) (2019) 220102.
[39] C. Zhao, K. Chen, Z. Wei, Y. Chen, D. Miao, W. Wang, Multilevel triplet deep learning model for person re-identification, Pattern Recognit. Lett. 117 (2019) 161–168.
[40] F. Cakir, K. He, X. Xia, B. Kulis, S. Sclaroff, Deep metric learning to rank, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1861–1870.
[41] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, arXiv:1806.01261 (2018).
[42] W. Li, J. Xu, J. Huo, L. Wang, Y. Gao, J. Luo, Distribution consistency based covariance metric networks for few-shot learning, AAAI, 2019.
[43] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[45] 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 (2018).
[46] G.E. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, CoRR (2015) abs/1503.02531.

[47] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, K. He, Data distillation: towards omni-supervised learning, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 4119–4128.

[48] T. Wang, J.-Y. Zhu, A. Torralba, A.A. Efros, Dataset distillation, CoRR (2018) abs/1811.10959.

[49] X. Zhang, F.X. Yu, S. Karaman, W. Zhang, S.-F. Chang, Heated-up softmax embedding, arXiv:1809.04157 (2018).

[50] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[51] A. Katharopoulos, F. Fleuret, Not all samples are created equal: deep learning with importance sampling, arXiv:1803.00942 (2018).

[52] B. Harwood, B.G. Kumar, G. Carneiro, I. Reid, T. Drummond, et al., Smart mining for deep metric learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2821–2829.

[53] Y. Zhao, Z. Jin, G.-j. Qi, H. Lu, X.-s. Hua, An adversarial approach to hard triplet generation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 501–517.

[54] A.R. Zamir, A. Sax, W. Shen, L.J. Guibas, J. Malik, S. Savarese, Taskonomy: disentangling task transfer learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3712–3722.

[55] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.

[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: a large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[57] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization, in: First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, 2011.

[58] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

[59] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, J.-B. Huang, A closer look at few-shot classification, in: International Conference on Learning Representations, 2019.

[60] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167 (2015).

Wei Zhu received the M.Sc. degrees in computer science from the Northwestern Polytechnical University, under the supervision of Prof. Feiping Nie, and 2018. He is currently pursuing the PhD degree with the Department of Computer Science, University of Rochester, under the supervision of Prof. Jiebo Luo. His research interests include computer vision and machine learning.

Wenbin Li is a Ph.D. student in the Department of Computer Science and Technology, Nanjing University. His current research interests include machine learning and computer vision.

Haofu Liao is a Ph.D. student in the Department of Computer Science, University of Rochester. His current research interests include machine learning and computer vision.

Jiebo Luo (S93, M96, SM99, F09) joined the Department of Computer Science at the University of Rochester in 2011, after a prolific career of over 15 years with Kodak Research. He has authored over 400 technical papers and holds over 90 U.S. patents. His research interests include computer vision, machine learning, data mining, social media,and biomedical informatics. He has served as the Program Chair of the ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, and on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BIG DATA, Pattern Recognition, Machine Vision and Applications, and ACM Transactions on Intelligent Systems and Technology. He is also a Fellow of ACM, AAAI, SPIE and IAPR.