Revisiting Local Descriptor based Image-to-Class Measure for Few-shot Learning

Wenbin Li[1], Lei Wang[2], Jinglin Xu[3], Jing Huo[1], Yang Gao[1], Jiebo Luo[4]
[1]Nanjing University, China, [2]University of Wollongong, Australia
[3]Northwestern Polytechnical University, China, [4]University of Rochester, USA

Abstract

Few-shot learning in image classification aims to learn a classifier to classify images when only few training examples are available for each class. Recent work has achieved promising classification performance, where an image-level feature based measure is usually used. In this paper, we argue that a measure at such a level may not be effective enough in light of the scarcity of examples in few-shot learning. Instead, we think a local descriptor based image-to-class measure should be taken, inspired by its surprising success in the heydays of local invariant features. Specifically, building upon the recent episodic training mechanism, we propose a Deep Nearest Neighbor Neural Network (DN4 in short) and train it in an end-to-end manner. Its key difference from the literature is the replacement of the image-level feature based measure in the final layer by a local descriptor based image-to-class measure. This measure is conducted online via a k-nearest neighbor search over the deep local descriptors of convolutional feature maps. The proposed DN4 not only learns the optimal deep local descriptors for the image-to-class measure, but also utilizes the higher efficiency of such a measure in the case of example scarcity, thanks to the exchangeability of visual patterns across the images in the same class. Our work leads to a simple, effective, and computationally efficient framework for few-shot learning. Experimental study on benchmark datasets consistently shows its superiority over the related state-of-the-art, with the largest absolute improvement of 17% *over the next best. The source code can be available from* `https://github.com/WenbinLee/DN4.git`.

1. Introduction

Few-shot learning aims to learn a model with good generalization capability such that it can be readily adapted to new unseen classes (concepts) by accessing only one or few examples. However, the extremely limited number of examples per class can hardly represent the class distribution effectively, making this task truly challenging.

To tackle the few-shot learning task, a variety of methods have been proposed, which can be roughly divided into two types, *i.e.,* meta-learning based [16, 14, 13] and metric-learning based [9, 17, 25]. The former type introduces a meta-learning paradigm [18, 21] to learn an across-task meta-learner for generalizing to new unseen tasks. They usually resort to recurrent neural networks or long short term memory networks to learn a memory network [24, 12] to store knowledge. The latter type adopts a relatively simpler architecture to learn a deep embedding space to transfer representation (knowledge). This type of methods usually relies on the metric learning and episodic training mechanism [22]. Both types of methods have greatly advanced the development of few-shot learning.

These existing methods mainly focus on making knowledge transfer [22, 2], concept representation [17, 4] or relation measure [25], but have not paid sufficient attention to the way of the final classification. They generally take the common practice, *i.e.,* using the image-level pooled features or fully connected layers designed for large-scale image classification, for the few-shot case. Considering the unique characteristic of few-shot learning (*i.e.,* the scarcity of examples for each training class), such a common practice may not be appropriate anymore.

In this paper, we revisit the Naive-Bayes Nearest-Neighbor (NBNN) approach [1] published a decade ago, and investigate its effectiveness in the context of the latest few-shot learning research. The NBNN approach demonstrated a surprising success when the bag-of-features model with local invariant features (*i.e.,* SIFT) was popular. That work provides two key insights. First, summarizing the local features of an image into a compact image-level representation could lose considerable discriminative information. It will not be recoverable when the number of training examples is small. Second, in this case, directly using these local features for classification will not work if an image-to-image measure is used. Instead, an image-to-class measure

should be taken, by exploiting the fact that a new image can be roughly "composed" using the pieces of other images in the same class. The above two insights inspire us to review the way of the final classification in the existing methods for few-shot learning and reconsider the NBNN approach for this task with deep learning.

Specifically, we develop a novel *Deep Nearest Neighbor Neural Network* (DN4 in short) for few-shot learning. It follows the recent episodic training mechanism and is fully end-to-end trainable. Its key difference from the related existing methods lies in that it replaces the image-level feature based measure in the final layer with a local descriptor based image-to-class measure. Similar to NBNN [1], this measure is computed via a k-nearest neighbor search over local descriptors, with the difference that these descriptors are now trained deeply via convolutional neural networks. Once trained, applying the proposed network to new few-shot learning tasks is straightforward, consisting of local descriptor extraction and then a nearest neighbor search. Interestingly, in terms of computation, the scarcity of examples per class now turns out to be an "advantage" making NBNN more appealing for few-shot learning. It mitigates the computation of searching for the nearest neighbors from a huge set of local descriptors, which is one factor of the lower popularity of NBNN in large-scale image classification.

Experiments are conducted on multiple benchmark datasets to compare the proposed DN4 with the original NBNN and the related state-of-the-art methods for the task of few-shot learning. The proposed method again demonstrates a surprising success. It improves the 1-shot and 5-shot accuracy on *mini*ImageNet from 50.44% to 51.24% and from 66.53% to 71.02%, respectively. Particularly, on fine-grained datasets, it achieves the largest absolute improvement over the next best method by 17%.

2. Related Work

Among the recent literature of few-shot learning, the transfer learning based methods are most relevant to the proposed method. Therefore, we briefly review two main branches of this kind of methods as follows.

Meta-learning based methods. As shown by the representative work [16, 14, 3, 2, 5], the meta-learning based methods train a meta-learner with the meta-learning or the learning-to-learn paradigm [18, 19, 21] for few-shot learning. This is beneficial for identifying how to update the parameters of the learner's model. For instance, Santoro *et al.* [16] trained an LSTM as a controller to interact with an external memory module. And the work [14] adopted an LSTM-based meta-learner as an optimizer to train another classifier as well as learning a task-common initialization for this classifier. The work of MM-Net [2] constructed a contextual learner to predict the parameters of an embedding network for unlabeled images by using memory slots.

Although the meta-learning based methods can achieve excellent results for few-shot classification, it is difficult to train their complicated memory-addressing architecture because of the temporally-linear hidden state dependency [13]. Compared with the methods in this branch, the proposed framework DN4 can be trained more easily in an end-to-end manner from scratch, *e.g.,* by only using a common single convolutional neural networks (CNN), and could provide quite competitive results.

Metric-learning based methods. The metric-learning based methods mainly depend on learning an informative similarity metric, as demonstrated by the representative work [9, 22, 20, 17, 4, 25, 11]. Specifically, to introduce the metric-based method into few-shot learning, Koch *et al.* [9] originally utilized a Siamese Neural Network to learn powerful discriminative representations and then generalized them to unseen classes. And then, Vinyals *et al.* [22] introduced the episodic training mechanism into few-shot learning and proposed the Matching Nets by combining attention and memory together. In [17], a Prototypical Network was proposed by taking the mean of each class as its corresponding prototype representation to learn a metric space. Recently, Sung *et al.* considered the relation between query images and class images, and presented a Relation Network [25] to learn a deep non-linear measure.

The proposed framework DN4 belongs to metric-learning based methods. However, a key difference from them is that the above methods mainly adopt the image-level features for classification, while the proposed DN4 exploits deep local descriptors and the image-to-class measure for classification, as inspired by the NBNN approach [1]. As will be shown in the experimental part, the proposed DN4 can clearly outperform the several state-of-the-art metric-learning based methods.

3. The Proposed Method

3.1. Problem Formulation

Let $\mathcal{S}$ denote a support set, which contains C different image classes and K labeled samples per class. Given a query set $\mathcal{Q}$, few-shot learning aims to classify each unlabeled sample in $\mathcal{Q}$ according to the set $\mathcal{S}$. This setting is also called C-way K-shot classification. Unfortunately, when $\mathcal{S}$ only has few samples per class, it will be hard to effectively learn a model to classify the samples in $\mathcal{Q}$. Usually, the literature resorts to an auxiliary set $\mathcal{A}$ to learn transferable knowledge to improve the classification on $\mathcal{Q}$. Note that the set $\mathcal{A}$ can contain a large number of classes and labeled samples, but it has a *disjoint* class label space with respect to the set $\mathcal{S}$.

The episodic training mechanism [22] has been demonstrated in the literature as an effective approach to learning the transferable knowledge from $\mathcal{A}$, and it will also

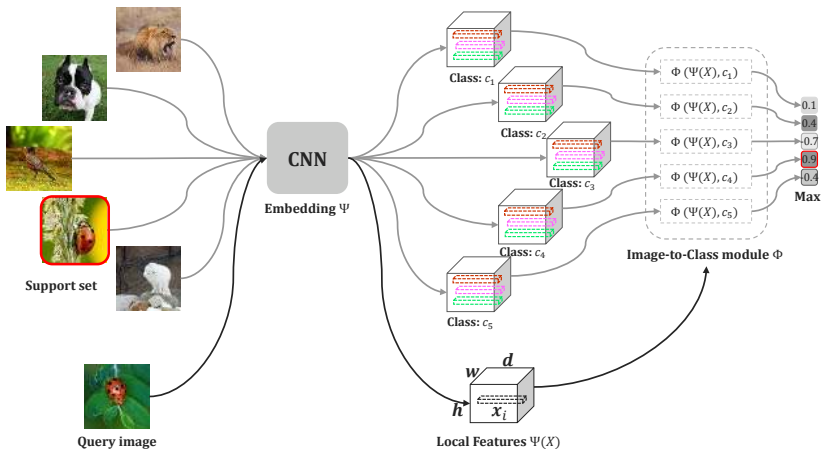


Figure 1. Illustration of the proposed Deep Nearest Neighbor Neural Network (DN4 in short) for a few-shot learning task in the 5-way and 1-shot setting. As shown, this framework consists of a CNN-based embedding module $\Psi(\cdot)$ for learning deep local descriptors and an image-to-class module $\Phi(\cdot)$ for measuring the similarity between a given query image X and each of the classes, c_i $(i = 1, 2, \cdots, 5)$.

be adopted in this work. Specifically, at each iteration, an episode is constructed to train the classification model by simulating a few-shot learning task. The episode consists of a support set $\mathcal{A}_S$ and a query set $\mathcal{A}_Q$ that are randomly sampled from the auxiliary set $\mathcal{A}$. Generally, $\mathcal{A}_S$ has the same numbers of ways (*i.e.,* classes) and shots as $\mathcal{S}$. In other words, there are exactly C classes and K samples per class in $\mathcal{A}_S$. During training, tens of thousands of episodes will be constructed to train the classification model, namely the episodic training. In the test stage, with the support set $\mathcal{S}$, the learned model can be directly used to classify each image in $\mathcal{Q}$.

3.2. Motivation from the NBNN Approach

This work is largely inspired by the Naive-Bayes Nearest-Neighbor (NBNN) method in [1]. The two key observations of NBNN are described as follows, and we show that they apply squarely to few-shot learning.

First, for the (then-popular) bag-of-features model in image classification, local invariant features are usually quantized into visual words to generate the distribution of words (*e.g.*, a histogram obtained by sum-pooling) in an image. It is observed in [1] that due to quantization error, such an image-level representation could significantly lose discriminative information. If there are sufficient training samples, the subsequent learning process (*e.g.*, via support vector machines) can somehow recover from such a loss, still showing satisfactory classification performance. Nevertheless, when training samples are insufficient, this loss is unrecoverable and leads to poor classification.

Few-shot learning is impacted more significantly by the issue of example scarcity than NBNN. And the existing methods usually pool the last convolutional feature maps (*e.g.*, via the global average pooling or fully connected layer) to an image-level representation for the final classification. In this case, such an information loss will also occur and is unrecoverable.

Second, as further observed in [1], using the local invariant features of two images, instead of their image-level representations, to measure an image-to-image similarity for classification will still incur a poor result. This is because such an image-to-image similarity does not generalize beyond training samples. When the number of training samples is small, a query image could be different from any training samples of the same class due to intra-class variation or background clutter. Instead, an image-to-class measure should be used. Specifically, the local invariant features from all training samples in the same class are collected into one pool. This measure evaluates the proximity (*e.g.,* via nearest-neighbor search) of the local features of a query image to the pool of each class for classification.

Again, this observation applies to few-shot learning. Essentially, the above image-to-class measure breaks the boundaries of training images in the same class, and uses their local features collectively to provide a richer and more flexible representation for a class. As indicated in [1], this setting can be justified by a fact that a new image can be roughly "composed" by using the pieces of other images in the same class (*i.e.*, the exchangeability of visual patterns across the images in the same class).

3.3. The Proposed DN4 Framework

The above analysis motivates us to review the way of the final classification in few-shot learning and reconsider the NBNN approach. This leads to the proposed framework *Deep Nearest Neighbor Neural Network* (DN4 in short).

As illustrated in Figure 1, DN4 mainly consists of two components: a deep embedding module Ψ and an image-to-class measure module Φ. The former learns deep local descriptors for all images. With the learned descriptors,

the latter calculates the aforementioned image-to-class measure. Importantly, these two modules are integrated into a unified network and trained in an end-to-end manner from scratch. Also, note that the designed image-to-class module can readily work with any deep embedding module.

Deep embedding module. The module Ψ routinely learns the feature representations for query and support images. Any proper CNN can be used. Note that Ψ only contains convolutional layers but has no fully connected layer, since we just need deep local descriptors to compute the image-to-class measure. In short, given an image X, $\Psi(X)$ will be an $h \times w \times d$ tensor, which can be viewed as a set of m ($m = hw$) d-dimensional local descriptors as

$$\Psi(X) = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{d \times m}\,, \tag{1}$$

where $\boldsymbol{x}_i$ is the i-th deep local descriptor. In our experiments, given an image with a resolution of 84×84, we can get $h = w = 21$ and $d = 64$. It means that each image has 441 deep local descriptors in total.

Image-to-Class module. The module Φ uses the deep local descriptors from all training images in a class to construct a local descriptor space for this class. In this space, we calculate the image-to-class similarity (or distance) between a query image and this class via k-NN, as in [1].

Specifically, through the module Ψ, a given query image q will be embedded as $\Psi(q) = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{d \times m}$. For each descriptor $\boldsymbol{x}_i$, we find its k-nearest neighbors $\hat{\boldsymbol{x}}_i^j|_{j=1}^k$ in a class c. Then we calculate the similarity between $\boldsymbol{x}_i$ and each $\hat{\boldsymbol{x}}_i$, and sum the mk similarities as the image-to-class similarity between q and the class c. Mathematically, the image-to-class measure can be easily expressed as

$$\begin{aligned} \Phi\big(\Psi(q), c\big) &= \sum_{i=1}^{m} \sum_{j=1}^{k} \cos(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i^j) \\ \cos(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i) &= \frac{\boldsymbol{x}_i^\top \hat{\boldsymbol{x}}_i}{\|\boldsymbol{x}_i\| \cdot \|\hat{\boldsymbol{x}}_i\|}\,, \end{aligned} \tag{2}$$

where $\cos(\cdot)$ indicates the cosine similarity. Other similarity or distance functions can certainly be employed.

Note that in terms of computational efficiency, the image-to-class measure seems more suitable for few-shot classification than the generic image classification focused in [1]. The major computational issue in NBNN caused by searching for k-nearest neighbors from a huge pool of local descriptors has now been substantially weakened due to the much smaller number of training samples in the few-shot setting. This makes the proposed framework computationally efficient. Furthermore, compared with NBNN, it will be more promising, by benefiting from the deep feature representations that are much more powerful than the hand-crafted features used in NBNN.

Finally, it is worth mentioning that the image-to-class module in DN4 is non-parametric. So the entire classification model is non-parametric if not considering the embedding module Ψ. Since a non-parametric model does not involve parameter learning, the over-fitting issue in parametric few-shot learning methods (*e.g.,* learning a fully connected layer over image-level representation) can also be mitigated to some extent.

3.4. Network Architecture

For a fair comparison with the state-of-the-art methods, we take a commonly used four-layer convolutional neural network as the embedding module. It contains four convolutional blocks, each of which consists of a convolutional layer, a batch normalization layer and a Leaky ReLU layer. Besides, for the first two convolutional blocks, an additional 2×2 max-pooling layer is also appended, respectively. This embedding network is named *Conv-64F*, since there are 64 filters of size 3×3 in each convolutional layer. As for the image-to-class module, the only hyper-parameter is the parameter k, which will be discussed in the experiment.

At each iteration of the episodic training, we feed a support set $\mathcal{S}$ and a query image q into our model. Through the embedding module Ψ, we obtain all the deep local representations for all these images. Then via the module Φ, we calculate the image-to-class similarity between q and each class by Eq. (2). For a C-way K-shot task, we can get a similarity vector $\boldsymbol{z} \in \mathbb{R}^C$. The class corresponding to the largest component of $\boldsymbol{z}$ will be the prediction for q.

4. Experimental Results

The main goal of this section is to investigate two interesting questions: (1) How does the pre-trained deep features based NBNN without episodic training perform on the few-shot learning? (2) How does our proposed DN4 framework, *i.e.,* a CNN based NBNN in an end-to-end episodic training manner, perform on the few-shot learning?

4.1. Datasets

We conduct all the experiments on four benchmark datasets as follows.

***mini*ImageNet.** As a mini-version of ImageNet [15], this dataset [22] contains 100 classes with 600 images per class, and has a resolution of 84×84 for each image. Following the splits used in [14], we take 64, 16 and 20 classes for training (auxiliary), validation and test, respectively.

Stanford Dogs. This dataset [7] is originally used for the task of fine-grained image classification, including 120 breeds (classes) of dogs with a total number of $20,580$ images. Here, we conduct fine-grained few-shot classification task on this dataset and take 70, 20 and 30 classes for training (auxiliary), validation and test, respectively.

Stanford Cars. This dataset [10] is also a benchmark dataset for fine-grained classification task, which consists of

196 classes of cars with a total number of $16,185$ images. Similarly, 130, 17 and 49 classes in this dataset are split for training (auxiliary), validation and test.

CUB-200. This dataset [23] contains 6033 images from 200 bird species. In a similar way, we select 130, 20 and 50 classes for training (auxiliary), validation and test.

For the last three fine-grained datasets, all the images in these datasets are resized to 84×84 as *mini*ImageNet.

4.2. Experimental Setting

All experiments are conducted around the C-way K-shot classification task on the above datasets. To be specific, 5-way 1-shot and 5-shot classification tasks will be conducted on all these datasets. During training, we randomly sample and construct $300,000$ episodes to train all of our models by employing the episodic training mechanism. In each episode, besides the K support images (shots) in each class, 15 and 10 query images will also be selected from each class for the 1-shot and 5-shot settings, respectively. In other words, for a 5-way 1-shot task, there will be 5 support images and 75 query images in one training episode. To train our model, we adopt Adam algorithm [8] with an initial learning rate of 1×10^{-3} and reduce it by half of every $100,000$ episodes.

During test, we randomly sample 600 episodes from the test set, and take the top-1 mean accuracy as the evaluation criterion. This process will be repeated five times, and the final mean accuracy will be reported. Moreover, the 95% confidence intervals are also reported. Notably, all of our models are trained from scratch in an end-to-end manner and do not need fine-tuning in the test stage.

4.3. Comparison Methods

Baseline methods. To illustrate the basic classification performance on the above datasets, we implement a baseline method *k-NN (Deep global features)*. Particularly, we adopt the basic embedding network *Conv-64F* and append three additional FC layers to train a classification network on the corresponding training (auxiliary) dataset. During test, we use this pre-trained network to extract features from the last FC layer and use a k-NN classifier to get the final classification results. Also, to answer the first question at the beginning of Section 4, we re-implement the NBNN algorithm [1] by using the pre-trained Conv-64F truncated from the above k-NN (Deep global features) method. This new NBNN algorithm employing the deep local descriptors instead of the hand-crafted descriptors (*i.e.,* SIFT), is called *NBNN (Deep local features)*.

Metric-learning based methods. As our method belongs to the metric-learning branch, we mainly compare our model with four state-of-the-art metric-learning based models, including *Matching Nets FCE* [22], *Prototypical Nets* [17], *Relation Net* [25] and *Graph Neural Network* (GNN) [4]. Note that we re-run the GNN model by using the Conv-64F as its embedding module because the original GNN adopts a different embedding module *Conv-256F*, which also has four convolutional layers but with 64, 96, 128 and 256 filters for the corresponding layers, respectively. Also, we re-run the Prototypical Nets via the same 5-way training setting instead of the 20-way training setting in the original work for a fair comparison.

Meta-learning based methods. Besides the metric-learning based models, five state-of-the-art meta-learning based models are also picked for reference. These models include *Meta-Learner LSTM* [14], *Model-agnostic Meta-learning (MAML)* [3], *Simple Neural AttentIve Learner (SNAIL)* [13], *MM-Net* [2] and *Dynamic-Net* [5]. As SNAIL adopts a much more complicated *ResNet-256F* (a smaller version of ResNet [6]) as its embedding module, we will additionally report its results based on the *Conv-32F* provided in its appendix for a fair comparison. Note that Conv-32F has the same architecture with Conv-64F, but with 32 filters per convolutional layer, which has also been employed by Meta-Learner LSTM and MAML to reduce over-fitting.

4.4. Few-shot Classification

The generic few-shot classification task is conducted on *mini*ImageNet. The results are reported in Table 1, where the hyper-parameter k is set as 3. From Table 1, it is amazing to see that NBNN (Deep local features) can achieve much better results than k-NN (Deep global features), and it is even better than Matching Nets FCE, Meta-Learner LSTM, and SNAIL (Conv-32F). This not only verifies that the local descriptors can perform better than the image-level features (*i.e.,* FC layer features used by k-NN), but also shows that the image-to-class measure is truly promising. However, NBNN (Deep local features) still has a large performance gap compared with the state-of-the-art Prototypical Nets, Relation Net and GNN. The reason is that, as a lazy learning algorithm, NBNN (Deep local features) does not have a training stage and also lacks the episodic training. So far, the first question has been answered.

On the contrary, our proposed DN4 embeds the image-to-class measure into a deep neural network, and can learn the deep local descriptors jointly by employing the episodic training, which indeed obtains superior results. Compared with the metric-learning based models, our DN4 (Conv-64F) gains 7.68%, 2.22%, 2.79% and 0.8% improvements over Matching Nets FCE, GNN‡ (Conv-64F), Prototypical Nets‡ (*i.e.,* via 5-way training setting) and Relation Net on the 5-way 1-shot classification task, respectively. On the 5-way 5-shot classification task, we can even get 15.71%, 7.52%, 4.49% and 5.7% significant improvements over these models. The reason is that these methods usually use image-level features whose number is too small, while our DN4 adopts learnable deep local descriptors which are

Table 1. The mean accuracies of the 5-way 1-shot and 5-shot tasks on the *mini*ImageNet dataset, with 95% confidence intervals. The second column refers to which kind of embedding module is employed, *e.g., Conv-32F* and *Conv-64F etc*. The third column denotes the type of this method, *i.e.,* meta-learning based or metric-learning based. * Results reported by the original work. ‡ Results re-implemented in the same setting for a fair comparison.

Model	Embedding	Type	5-Way Accuracy (%)	
			1-shot	5-shot
k-NN (Deep global features)	*Conv-64F*	Metric	27.23±1.41	49.29±1.56
NBNN (Deep local features)	*Conv-64F*	Metric	44.10±1.17	58.84±1.10
Matching Nets FCE* [22]	*Conv-64F*	Metric	43.56±0.84	55.31±0.73
Prototypical Nets‡ [17]	*Conv-64F*	Metric	48.45±0.96	66.53±0.51
Prototypical Nets* [17]	*Conv-64F*	Metric	49.42±0.78	68.20±0.66
GNN‡ [4]	*Conv-64F*	Metric	49.02±0.98	63.50±0.84
GNN* [4]	*Conv-256F*	Metric	50.33±0.36	66.41±0.63
Relation Net* [25]	*Conv-64F*	Metric	50.44±0.82	65.32±0.70
Our DN4 (k=3)	*Conv-64F*	Metric	**51.24±0.74**	**71.02±0.64**
To take a whole picture of the-state-of-art methods				
Meta-Learner LSTM* [14]	*Conv-32F*	Meta	43.44±0.77	60.60±0.71
SNAIL* [13]	*Conv-32F*	Meta	45.10	55.20
MAML* [3]	*Conv-32F*	Meta	48.70±1.84	63.11±0.92
MM-Net* [2]	*Conv-64F*	Meta	53.37±0.48	66.97±0.35
SNAIL* [13]	*ResNet-256F*	Meta	55.71±0.99	68.88±0.92
Dynamic-Net* [5]	*ResNet-256F*	Meta	55.45±0.89	70.13±0.68
Dynamic-Net* [5]	*Conv-64F*	Meta	56.20±0.86	72.81±0.62

Table 2. The mean accuracies of the 5-way 1-shot and 5-shot tasks on three fine-grained datasets, *i.e., Stanford Dogs, Stanford Cars and CUB-200*, with 95% confidence intervals. For each setting, the best and the second best methods are highlighted.

Model	Embed.	5-Way Accuracy (%)					
		Stanford Dogs		*Stanford Cars*		*CUB-200*	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
k-NN (Deep global features)	*Conv-64F*	26.14±0.91	43.14±1.02	23.50±0.88	34.45±0.98	25.81±0.90	45.34±1.03
NBNN (Deep local features)	*Conv-64F*	31.42±1.12	42.17±0.99	28.18±1.24	38.27±0.92	35.29±1.03	47.97±0.96
Matching Nets FCE‡ [22]	*Conv-64F*	35.80±0.99	47.50±1.03	34.80±0.98	44.70±1.03	45.30±1.03	59.50±1.01
Prototypical Nets‡ [17]	*Conv-64F*	37.59±1.00	48.19±1.03	40.90±1.01	52.93±1.03	37.36±1.00	45.28±1.03
GNN‡ [4]	*Conv-64F*	**46.98±0.98**	62.27±0.95	55.85±0.97	71.25±0.89	**51.83±0.98**	63.69±0.94
Our DN4 (k=1)	*Conv-64F*	45.41±0.76	**63.51±0.62**	**59.84±0.80**	**88.65±0.44**	46.84±0.81	**74.92±0.64**
Our DN4-DA (k=1)	*Conv-64F*	**45.73±0.76**	**66.33±0.66**	**61.51±0.85**	**89.60±0.44**	**53.15±0.84**	**81.90±0.60**

more abundant especially in the 5-shot setting. On the other hand, local descriptors enjoy the exchangeability characteristic, making the distribution of each class built upon the local descriptors more effective than the one built upon the image-level features. Therefore, the second question can also be answered.

To take a whole picture of the few-shot learning area, we also report the results of the state-of-the-art meta-learning based methods. We can see that our DN4 is still competitive with these methods. Especially in the 5-way 5-shot setting, our DN4 gains 15.82%, 10.42%, 7.91% and 4.05% improvements over SNAIL (Conv-32F), Meta-Learner LSTM, MAML and MM-Net, respectively. As for the Dynamic-Net, a two-stage model, it pre-trains its model with all classes together before conducting the few-shot training, while our DN4 does not. More importantly, our DN4 only has one single unified network, which is much simpler than these meta-learning based methods with additional complicated memory-addressing architectures.

4.5. Fine-grained Few-shot Classification

Besides the generic few-shot classification, we also conduct fine-grained few-shot classification tasks on three fine-grained datasets, *i.e.,* Stanford Dogs, Stanford Cars and CUB-200. Two baseline models and three state-of-the-art models are implemented on these three datasets, *i.e.,*

k-NN (Deep global features), NBNN (Deep local features), Matching Nets FCE [22], Prototypical Nets [17] and GNN [4]. The results are shown in Table 2. In general, the fine-grained few-shot classification task is more challenging than the generic one due to the smaller inter-class and larger intra-class variations of the fine-grained datasets. It can be seen by comparing the performance of the same methods between Tables 1 and 2. The performance of the k-NN (Deep global features), NBNN (Deep local features) and Prototypical Nets on the fine-grained datasets is worse than that on *mini*ImageNet. It can also be observed that NBNN (Deep local features) performs consistently better than k-NN (Deep global features).

Due to the small inter-class variation of the fine-grained task, we choose $k = 1$ for our DN4 to avoid introducing noisy visual patterns. From Table 2, we can see that our DN4 performs surprisingly well on these datasets under the 5-shot setting. Especially on the Stanford Cars, our DN4 gains the largest absolute improvement over the second best method, *i.e.,* GNN, by 17%. Under the 1-shot setting, our DN4 does not perform as well as in the 5-shot setting. The key reason is that our model relies on the k-nearest neighbor algorithm, which is a lazy learning algorithm and its performance depends largely on the number of samples. This characteristic has been shown in Table 5, *i.e.,* the performance of DN4 gets better and better as the number of shots increases. Another reason is that these fine-grained datasets are not sufficiently large (*e.g.,* CUB-200 only has 6033 images), resulting in over-fitting when training deep networks.

To avoid over-fitting, we perform data augmentation on the training (auxiliary) sets by cropping and horizontally flipping randomly. Then, we re-train our model, *i.e.,* DN4-DA, on these augmented datasets but test on the original test sets. It can be observed that our DN4-DA can obtain nearly the best results for both 1-shot and 5-shot tasks. The fine-grained recognition largely relies on the subtle local visual patterns, and they can be naturally captured by the learnable deep local descriptors emphasized in our model.

4.6. Discussion

Ablation study. To further verify that the image-to-class measure is more effective than the image-to-image measure, we perform an ablation study by developing two image-to-image (IoI for short) variants of DN4. Specifically, the first variant named DN4-IoI-1 concatenates all local descriptors of an image as a high-dimensional ($h \times w \times d$) feature vector and uses the image-to-image measure. As for the second variant (DN4-IoI-2 for short), it keeps the local descriptors like DN4 without concatenation. The only difference between DN4-IoI-2 and DN4 is that DN4-IoI-2 restricts the search for the k-NN of a query's local descriptor within each individual support image, while DN4 can search from one entire support class. Under the 1-shot setting, DN4-IoI-2 is identical with DN4. Both variants still adopt the k-NN search, and use $k = 1$ and $k = 3$ for 1-shot setting and 5-shot setting, respectively.

Table 3. The results of the ablation study on *mini*ImageNet.

Model	5-Way Accuracy (%)	
	1-shot	5-shot
DN4-IoI-1	37.39 ± 0.82	50.47 ± 0.66
DN4-IoI-2	51.14 ± 0.79	69.52 ± 0.62
DN4	$\mathbf{51.24\pm0.74}$	$\mathbf{71.02\pm0.64}$

Table 4. The 5-way 5-shot mean accuracy (%) of our DN4 by varying the value of $k \in \{1, 3, 5, 7\}$ during training on *mini*ImageNet.

Model	5-way 5-shot Accuracy (%)			
	$k = 1$	$k = 3$	$k = 5$	$k = 7$
DN4	**71.95**	71.02	70.20	68.56

The results on *mini*ImageNet are reported in Table 3. As seen, DN4-IoI-1 performs clearly the worst by using the concatenated global features with the image-to-image measure. In contrast, DN4-IoI-2 performs excellently on both 1-shot and 5-shot tasks, which verifies the importance of local descriptors and the exchangeability (within one image). Notably, DN4 is superior to DN4-IoI-2 on the 5-shot task, which shows that utilizing the exchangeability of visual patterns within a class indeed helps to gain performance.

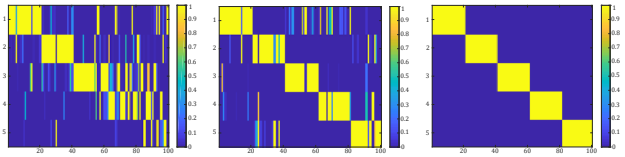
Influence of backbone networks. Besides the commonly used *Conv-64F*, we also evaluate our model by using another deeper embedding module, *i.e., ResNet-256F* used by SNAIL [13] and Dynamic-Net [5]. The details of ResNet-256F can refer to SNAIL [13]. When using ResNet-256F as the embedding module, the accuracy of DN4 reaches $52.44 \pm 0.80\%$ for the 5-way 1-shot task and $72.53 \pm 0.62\%$ for the 5-shot task. As seen, with a deeper backbone network, DN4 can perform better than the case of using the shallow Conv-64F. Moreover, when using the same ResNet-256F as the embedding module, our DN4 (ResNet-256F) can gain 2.4% improvements over Dynamic-Net (ResNet-256F) (*i.e.,* $70.13 \pm 0.68\%$) under the 5-shot setting (see Table 1).

Influence of neighbors. In the image-to-class module, we need to find the k-nearest neighbors in one support class for each local descriptor of a query image. Next, we measure the image-to-class similarity between a query image and a specific class. How to choose a suitable hyper-parameter k is thus a key. For this purpose, we perform a 5-way 5-shot task on *mini*ImageNet by varying the value of $k \in \{1, 3, 5, 7\}$, and show the results in Table 4. It can be seen that the value of k has a mild impact on performance. Therefore, in our model, k should be selected according to the specific task.

Influence of shots. The episodic training mechanism is

Table 5. The 5-way K-shot mean accuracy (%) of our DN4 by varying the number of shots ($K = 1, 2, 3, 4, 5$) during training on *mini*ImageNet. For each test setting, the best result is highlighted.

Train	Test				
	1-shot	2-shot	3-shot	4-shot	5-shot
1-shot	51.24	58.13	62.10	64.22	66.10
2-shot	50.69	58.53	62.31	64.84	66.49
3-shot	53.22	60.74	64.95	67.52	69.35
4-shot	52.43	60.90	65.33	67.93	69.70
5-shot	**53.85**	**61.78**	**66.16**	**68.92**	**71.02**



(a) NBNN (b) Our DN4 (c) Ground Truth

Figure 2. Similarity matrices of NBNN (Deep local Features), our DN4 and the ground truth on *mini*ImageNet under the 5-way 5-shot setting. Vertical axis denotes the five classes in the support set. Horizontal axis denotes 20 query images per class. The warmer colors indicate higher similarities.

popular in current few-shot learning methods. The basic rule is the matching condition between training and test. It means that, in the training stage, the numbers of ways and shots should keep consistent with those adopted in the test stage. In other words, if we want to perform a 5-way 1-shot task, the same 5-way 1-shot setting should be maintained in the training stage. However, in the real training stage, we still want to know the influence of mismatching conditions, *i.e.,* under-matching condition and over-matching condition. We find that the over-matching condition can achieve better performance than the matching condition, and much better than the under-matching condition.

Basically, for the under-matching condition, we use a smaller number of shots in the training stage, and conversely, use a larger number of shots for the over-matching condition. We fix the number of ways but vary the number of shots during training to learn several different models. Then we test these models under different shot settings, where the number of shots is changed but the number of ways is fixed. A 5-way K-shot ($K = 1, 2, 3, 4, 5$) task is conducted on *mini*ImageNet by using our DN4. The results are presented in Table 5, where the entries on the diagonal are the results of the matching condition. The results in the upper triangle are the results of the under-matching condition. Also, the lower triangle contains the results of the over-matching condition. It can be seen that the results in the lower triangle are better than those on the diagonal, and the results on the diagonal are better than those in the upper triangle. This exactly verifies our statement made above. It is also worth mentioning that if we use a 5-shot trained model and test it on the 1-shot task, we can obtain an accuracy of 53.85%. This result is quite high in this task, and much better than 51.24% obtained by the 1-shot trained model using our DN4 under a matching condition.

Visualization. We visualize the similarity matrices learned by NBNN (Deep local features) and our DN4 under the 5-way 5-shot setting on *mini*ImageNet. Both of them are image-to-class measure based models. We select 20 query images from each class (*i.e.,* 100 query images in total), calculate the similarity between each query image and each class, and visualize the 5×100 similarity matrices. From Figure 2, it can be seen that the results of DN4 are much closer to the ground truth than those of NBNN, which demonstrates that the end-to-end manner is more effective.

Runtime. Although NBNN performs successfully in the literature [1], it did not become popular. One key reason is the high computational complexity of the nearest-neighbor search, especially in large-scale image classification tasks. Fortunately, under the few-shot setting, our framework can enjoy the excellent performance of NBNN without being significantly affected by its computational issue. Generally, during training for a 5-way 1-shot or 5-shot task, one episode (batch) time is 0.31s or 0.38s with 75 or 50 query images on a single Nvidia GTX 1080Ti GPU and a single Intel i7-3820 CPU. During test, it will be more efficient, and only takes 0.18s for one episode. Moreover, the efficiency of our model can be further improved with optimized parallel implementation.

5. Conclusions

In this paper, we revisit the local descriptor based image-to-class measure and propose a simple and effective Deep Nearest Neighbor Neural Network (DN4) for few-shot learning. We emphasize and verify the importance and value of the learnable deep local descriptors, which are more suitable than image-level features for the few-shot problem and can well boost the classification performance. We also verify that the image-to-class measure is superior to the image-to-image measure, owing to the exchangeability of visual patterns within a class.

Acknowledgements

This work is partially supported by the NSF awards (Nos. 1704309, 1722847, 1813709), National NSF of China (Nos. 61432008, 61806092), Jiangsu Natural Science Foundation (No. BK20180326), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (No. CX201814).

References

[1] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, pages 1–8. IEEE, 2008.

[2] Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei. Memory matching networks for one-shot image recognition. In *CVPR*, pages 4080–4088, 2018.

[3] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[4] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. *ICLR*, 2018.

[5] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop*, volume 2, page 1, 2011.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[9] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshop*, volume 2, 2015.

[10] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshop*, pages 554–561, 2013.

[11] W. Li, J. Xu, J. Huo, L. Wang, G. Yang, and J. Luo. Distribution consistency based covariance metric networks for few-shot learning. In *AAAI*, 2019.

[12] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. *EMNLP*, 2016.

[13] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. *ICLR*, 2018.

[14] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2017.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[16] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016.

[17] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4080–4090, 2017.

[18] S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

[19] S. Thrun and L. Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

[20] E. Triantafillou, R. Zemel, and R. Urtasun. Few-shot learning through an information retrieval lens. In *NIPS*, pages 2255–2265, 2017.

[21] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *AIR*, 18(2):77–95, 2002.

[22] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[23] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[24] J. Weston, S. Chopra, and A. Bordes. Memory networks. *ICLR*, 1410.3916, 2015.

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