Distribution Consistency based Covariance Metric Networks for Few-shot Learning

Wenbin Li[1*], Jinglin Xu[2*], Jing Huo[1], Lei Wang[3], Yang Gao[1], Jiebo Luo[4]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Northwestern Polytechnical University, China
[3]University of Wollongong, Australia
[4]University of Rochester, USA

Abstract

Few-shot learning aims to recognize new concepts from very few examples. However, most of the existing few-shot learning methods mainly concentrate on the first-order statistic of concept representation or a fixed metric on the relation between a sample and a concept. In this work, we propose a novel end-to-end deep architecture, named Covariance Metric Networks (CovaMNet). The CovaMNet is designed to exploit both the covariance representation and covariance metric based on the distribution consistency for the few-shot classification tasks. Specifically, we construct an embedded local covariance representation to extract the second-order statistic information of each concept and describe the underlying distribution of this concept. Upon the covariance representation, we further define a new deep covariance metric to measure the consistency of distributions between query samples and new concepts. Furthermore, we employ the episodic training mechanism to train the entire network in an end-to-end manner from scratch. Extensive experiments in two tasks, generic few-shot image classification and fine-grained few-shot image classification, demonstrate the superiority of the proposed CovaMNet. The source code can be available from `https://github.com/WenbinLee/CovaMNet.git`.

1 Introduction

Few-shot learning is a learning mechanism that tries to learn and understand new concepts (or categories) from only one or few examples. Humans can learn new concepts with very few instances, and have a strong generalization capability for their variants. Unfortunately, many current machine learning algorithms do not have such a strong generalization ability to identify a new category. Moreover, in real applications, new samples from new categories are usually difficult to obtain. It is even more difficult to make annotations in many applications. Therefore, learning new categories with very few samples becomes an urgent and important problem.

In recent years, a variety of methods have been proposed to handle this problem. One basic and straightforward way is to only utilize the information of a few examples. For example, the k-Nearest Neighbor classifier can be used to predict the label of the query image depending solely on the similarities between the query image and the few samples, without introducing any learning mechanism. Nevertheless, it is almost impossible to learn a real concept containing diverse and complex appearances merely by a few examples in this way. Therefore it is necessary to exploit more prior knowledge to precisely represent and learn.

Another commonly used approach is to resort to an additional auxiliary dataset via a transfer learning mechanism. More concretely, one emerging direction is to exploit meta-learning or learning-to-learn paradigm on the few-shot learning problem, such as (Santoro et al. 2016; Ravi and Larochelle 2017; Mishra et al. 2018). This kind of methods attempts to learn an across-task meta-learner which is trained on a distribution of similar tasks, aiming to generalize to unseen new tasks. Generally, a Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) network is employed to learn the meta-learner to capture the significant short- and long-term memory (knowledge) further. Furthermore, in order to tackle the problem of limited-data, another emerging research direction on metric-learning based methods tries to learn a deep embedding space by introducing metric-learning mechanism, *e.g.*, (Koch, Zemel, and Salakhutdinov 2015; Snell, Swersky, and Zemel 2017; Yang et al. 2018). Such methods usually utilize the idea of episodic training on the auxiliary dataset to learn transferable representations (knowledge).

Based on the above analysis, the key issue of few-shot learning is that the data of each category is too limited to express a concept adequately. Only using very few examples, either for training a learner or for fine-tuning a pre-trained model, can easily result in over-fitting. How to learn and store the transferable knowledge by fully utilizing the auxiliary data? How to represent a concept precisely in the few-shot setting? And how to measure the relation between a concept and a query sample reasonably? We will consider these three aspects to solve the problems occurred in the previous work, which are transferable knowledge, concept representation, and relation metric.

For the first aspect, meta-learning based methods employ meta-learner and the metric-learning based methods rely on the episodic training to capture the transferable knowledge, respectively. As for the proposed method, we adopt the episodic training mechanism because it is simpler than a

[*]Equal contribution

meta-learner but efficient. For the second aspect, Prototypical Nets (Snell, Swersky, and Zemel 2017) take the means of categories as prototypes to represent the concepts, and (Garcia and Bruna 2018) use a graph to represent a concept. In the proposed method, we exploit the second-order statistic of concept representation and verify that it is more suitable to represent a concept beyond the first-order statistic (*e.g.,* mean representation). For the third aspect, most of existing methods adopt the Euclidean distance (Snell, Swersky, and Zemel 2017) or cosine similarity (Vinyals et al. 2016) to measure the relation between a concept and a query sample, and then perform the classification. In addition, (Yang et al. 2018) trains a Relation Network by learning a deep non-linear metric to compare the relation between images. In the proposed method, we defined a novel deep *Covariance Metric*, which naturally captures the distribution consistency between a query sample and a concept.

Considering all the aforementioned three aspects, we design a *Covariance Metric Network* (CovaMNet) to deal with the problems of the few-shot learning. Specifically, the proposed CovaMNet introduces the episodic training mechanism to learn the transferable knowledge, which is simpler than Recurrent Networks, and also has been proved to be efficient. Moreover, we define a *Local Covariance Representation* and embed it into the network to learn each concept (or category). Because the covariance matrix exactly contains the second-order statistic information, it can naturally capture the underlying distribution information of each concept (or category) and thus becomes a good concept representation. Furthermore, we construct a *Covariance Metric Layer* based on the local covariance representation to measure the relation between a concept and a query sample by calculating the consistency between their distributions.

This paper designs a novel and effective framework based on a covariance metric, which considers three aspects (*i.e.*, transferable knowledge, concept representation and relation metric) and contains the local and global metric information, to solve the problem of few-shot learning. Formally, the contributions can be summarized as follows: (1) A novel and compact end-to-end Covariance Metric Network (CovaMNet) is proposed, aiming to address the above three aspects of few-shot learning. (2) We design a local covariance representation, which has the ability to represent a concept (or category) by utilizing a covariance matrix under the few-shot setting. (3) We construct a covariance metric to be the relation measure by calculating the distribution consistency between a query sample and each category. (4) Extensive experiments on several benchmark datasets demonstrate that our proposed framework shows its superiorities both in the generic few-shot classification and the fine-grained few-shot classification.

2 Related work

2.1 Transfer Learning Based Methods for Few-shot Learning

Recently, there are lots of previous work about the few-shot learning, where the transfer learning mechanism based methods are most relevant to our proposed method. Therefore, we will briefly review two main streams for this kind of methods, *i.e.*, the Meta-learning based methods and the Metric-learning based methods.

Meta-learning based methods: Meta-learning based methods introduce the meta-learning paradigm (Thrun 1998; Vilalta and Drissi 2002) or learning to learn (Thrun and Pratt 1998) into the few-shot learning. That is to train a meta-learner that learns how to update the parameters of the learner's model, referring to some representative methods (Santoro et al. 2016; Ravi and Larochelle 2017; Finn, Abbeel, and Levine 2017; Cai et al. 2018). For example, (Santoro et al. 2016) presented a memory-augmented model, where a LSTM was trained as a controller to interact with an external memory module. Also, (Ravi and Larochelle 2017) adopted a LSTM-based meta-learner as an optimizer to train another learner classifier as well as learning a task-common initialization for this classifier.

The meta-learning based methods are promising and achieve competitive results for few-shot classification. However, the complicated memory-addressing architecture (such as, RNN) used in these methods is difficult to train due to the temporally-linear hidden state dependency (Mishra et al. 2018). On the contrary, our proposed framework CovaMNet is only based on a single CNN, which can be trained easily in an end-to-end manner from scratch.

Metric-learning based methods: Metric learning based methods mainly rely on learning an informative similarity metric, including (Koch, Zemel, and Salakhutdinov 2015; Vinyals et al. 2016; Triantafillou, Zemel, and Urtasun 2017; Snell, Swersky, and Zemel 2017; Garcia and Bruna 2018; Yang et al. 2018). Typically, (Koch, Zemel, and Salakhutdinov 2015) was the first to introduce the metric-based method into one-shot learning, which adopted a Siamese Neural Network to learn powerful discriminative representations and then generalize to unseen categories. Later, (Vinyals et al. 2016) proposed the Matching Nets which combined attention and memory to enable rapid learning under the matched test and train conditions (*i.e.,* episodic training), obviating the fine-tuning to adapt to new categories.

There are two pieces of work closely related to ours. The first one, Prototypical Networks (Snell, Swersky, and Zemel 2017), learned a metric space and took the mean of each category as its corresponding prototype representation. Its classification was performed by calculating the distances between different prototype representations and then was generalized to new categories with very few examples. The second one, Relation Network (Yang et al. 2018), learned an embedding and a deep non-linear distance metric for comparing query and sample items. This network was trained end-to-end by utilizing the episodic training mechanism, which could tune the embedding and distance metric for effective few-shot learning.

Different from the above two methods, the proposed CovaMNet employs a more informative second-order statistic to express a concept, that is learning a covariance representation embedded in the network. This is because the covariance representation, as a second-order statistic, is more suitable to describe the underlying distribution of a concept. In addition, we design a novel deep covariance metric which does well in capturing the distribution consistency between query samples and categories. On the contrary, Prototypical Network merely adopted a fixed metric (*i.e.*, Euclidean distance) to measure the relation and perform the final classification. Relation Network learned a deep metric (*i.e.*, simply concatenating the feature maps between query and support samples) to obtain the classification results.

2.2 Covariance Pooling

Instead of the first-order pooling (*e.g.*, max-pooling) commonly used in the Convolutional Neural Networks (CNNs), the *Global Covariance Pooling* tries to explore a second-order pooling (covariance pooling) in CNNs. With the help of matrix square-root normalization, covariance pooling achieved impressive performance (Lin and Maji 2017). For example, (Li et al. 2017) proposed a Matrix Power Normalized Covariance (MPN-COV) method by producing covariance matrices over the last convolutional features to act as global image representations. However, MPN-COV needs eigenvalue decomposition (EIG) for the matrix square-root normalization, which suffered from high time complexity. Later, to accelerate the computation further, (Li et al. 2018) proposed an iterative matrix square-root normalization method for fast end-to-end training. In addition, (Wang, Li, and Zhang 2017) employed a Gaussian embedding strategy to explore new ways of inserting parametric probability distributions into the CNNs. Essentially, several bilinear pooling based methods (Lin, RoyChowdhury, and Maji 2015; Gao et al. 2016; Lin, RoyChowdhury, and Maji 2018) all belong to the covariance pooling based methods.

The covariance matrix is also utilized in our CovaMNet. The major differences between ours and the ones used in covariance pooling based methods are summarized as follows. Firstly, we utilize the covariance matrix in a different way. Our method takes the covariance matrix as a representation to capture the underlying distribution for a category (containing all images belonging to this category), whereas other covariance pooling based methods only use the covariance matrix to serve as a pooling method to obtain the global representation for every single image. Second, our method designs the covariance metric based on the raw covariance matrix to achieve the relation metric, which is more efficient. On the contrary, other covariance pooling methods perform the matrix square-root normalization depending on EIG or SVD, which is too computationally heavy to train.

3 The Proposed Method

3.1 Problem Formulation

Given three sets: a support set $\boldsymbol{S}$, a query set $\boldsymbol{Q}$ and an auxiliary set $\boldsymbol{A}$, the set $\boldsymbol{S}$ contains C different categories, each of which has K labeled samples. The set $\boldsymbol{Q}$ consists of unlabeled samples, which shares the same label space with the set $\boldsymbol{S}$. Different from the set $\boldsymbol{S}$, the set $\boldsymbol{A}$ contains lots of categories and labeled samples (much larger than C and K, respectively), which can be used for learning a mapping function and extracting transferable knowledge. It is worth noting that the label space of set $\boldsymbol{A}$ is disjoint with the label space of set $\boldsymbol{S}$.

Based on above definitions, the task of few-shot classification can be formally established as follows. The goal of few-shot classification is to perform the few-shot learning based on the set $\boldsymbol{S}$ and obtain the satisfactory classification results on the set $\boldsymbol{Q}$. Unfortunately, the set $\boldsymbol{S}$ with very few samples has no ability to learn an optimal mapping function to classify the set $\boldsymbol{Q}$, therefore we will resort to the auxiliary set and utilize an episodic training mechanism on this set, which has been verified to be effective in the work (Vinyals et al. 2016). Specifically, at each iteration, one episode is constructed by a subset which randomly samples C categories from the set $\boldsymbol{A}$. In this subset, the labeled samples of each category are randomly split into two sets: the support set $\boldsymbol{A}_S$ and the query set $\boldsymbol{A}_Q$, where K labeled samples in $\boldsymbol{A}_S$ and the rest in $\boldsymbol{A}_Q$. After t iterations, t episodes have been used to train the mapping function, namely the episodic training. Once trained, we predict the labels of the set $\boldsymbol{Q}$ via the mapping function conditioned on the set $\boldsymbol{S}$.

3.2 Model

The main contribution of our proposed method is to design a novel and compact end-to-end CovaMNet, whose two key components are the *Local Covariance Representation* and the *Covariance Metric*. They are embedded in two modules: a convolutional embedding module and a covariance metric module. The first module adopts an alternative CNN to learn rich image representations which are the bases of the local covariance representation. Under the few-shot setting, the local covariance representation is able to denote a concept (or category) utilizing a covariance matrix, owing to the second-order statistic property of the covariance matrix. The second module is constructed by a covariance metric layer, with the help of the first module, to measure the relation between a query sample and each category by calculating their distribution consistency. Most importantly, these two modules are integrated into a unified network and trained in an end-to-end manner from scratch. In this way, the representations and metrics can be learned simultaneously, making them complement and work best with each other.

Local Covariance Representation The covariance matrix has been widely used as a region descriptor (Tuzel, Porikli, and Meer 2006; 2007; Tabia et al. 2014) or a general representation (Wang et al. 2012; Harandi, Salzmann, and Porikli 2014; Wang et al. 2015b; 2015a; Huang et al. 2015; Wang et al. 2017), due to its favorable properties, *e.g.*, second-order statistic, symmetric positive-definiteness and so on.

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K] \in \mathbb{R}^{d \times K}$ be a data matrix and $\boldsymbol{x}_i \in \mathbb{R}^d$. The sample-based covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ can be defined as,

$$\boldsymbol{\Sigma} = \frac{1}{K-1} \sum_{i=1}^{K} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top , \qquad (1)$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean of K samples.

The covariance matrix is a raw second-order statistic of a sample set, which can describe the underlying distribution of this set directly. Therefore, employing the covariance matrix to represent the distribution of the few-shot categories

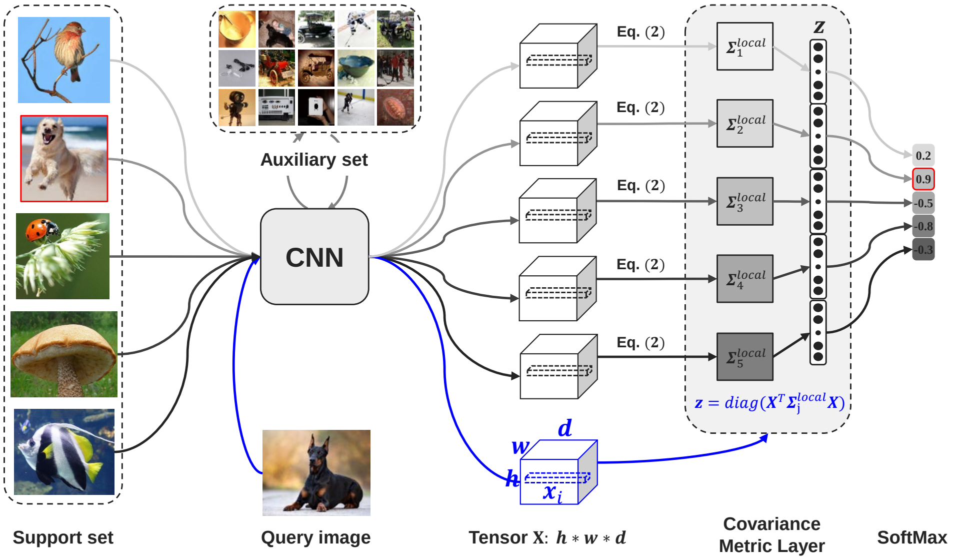

Figure 1: Architecture of the proposed CovaMNet for 5-way 1-shot task. Given a support set consisting of 5 categories with 1 image per category, a CNN (*i.e.,* embedding module) is learned from the auxiliary set to extract covariance representations for each category, followed by a covariance metric layer to measure the distribution consistency between a query image and each category. Finally, a softmax layer with the cross-entropy loss is exploited to get the classification result. The training of CovaMNet exactly matches the inference.

is a reasonable and promising way. However, unlike the covariance representation of a general image set (Wang et al. 2012), there are very few images (*e.g.*, $K = 1 \sim 5$) in each category under the few-shot setting, which makes it difficult to guarantee the nonsingular property of the covariance matrix. Besides, the number of samples in each category (*i.e.*, K) is too small to exactly learn a covariance matrix to describe the data distribution.

Hence, we combine all local deep descriptors of each category to calculate a local covariance representation for this category. Given an image set of the c-th category $\boldsymbol{D}_c = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K\}, \boldsymbol{X}_i \in \mathbb{R}^{d\times M}$, where $\boldsymbol{D}_c$ contains K images and each image $\boldsymbol{X}_i$ is represented by M local deep descriptors with d dimensions per descriptor. Therefore, the local covariance representation of the c-th category $\boldsymbol{\Sigma}_c^{local} \in \mathbb{R}^{d\times d}$ can be defined in a matrix form as below,

$$\boldsymbol{\Sigma}_c^{local} = \frac{1}{MK-1}\sum_{i=1}^{K}(\boldsymbol{X}_i - \boldsymbol{\tau})(\boldsymbol{X}_i - \boldsymbol{\tau})^\top , \qquad (2)$$

where $\boldsymbol{\tau} \in \mathbb{R}^{d\times M}$ is a matrix of mean vectors, with each of its column the same mean vector of all the MK descriptors.

Since the number of all local deep descriptors (*i.e.,* $441 \le MK \le 2205$) is much larger than the feature dimensionality (*i.e.,* $d = 64$), it can guarantee the non-singularity of the covariance matrix when calculating the covariance matrix. Another advantage is to capture the local detailed information of each category as well as facilitating the subsequent image recognition task, especially in the fine-grained image recognition. Furthermore, since this local covariance matrix is embedded into the deep network, it can be learned iteratively as the network updates, rather than fixed like pixel- or image-based covariance matrix. It is worth noting that the proposed model has no restriction on the number of shots, which means that the different number of shots can be adopted via a covariance representation during training and testing.

Covariance Metric To measure the relation between a sample and one category, a new measure function named as *Covariance Metric* is defined as follows:

$$f(\boldsymbol{x}, \boldsymbol{\Sigma}) = \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} \, , \qquad (3)$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is a sample with zero-mean over the query image and $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ indicates the covariance matrix of one specific category. The value of Eq. (3) will achieve a maximum based on the first k eigenvalues if $\boldsymbol{x}$ is in the direction of the first k eigenvectors of $\boldsymbol{\Sigma}$ according to Theorem 1. This means that the direction of $\boldsymbol{x}$ is in the major spread direction of this category and the distributions of $\boldsymbol{x}$ and this category are consistent.

Theorem 1. *Suppose that* $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ *is the covariance matrix of one specific category from the support set* $\boldsymbol{S}$*, satisfying* $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$*, where the diagonal matrix* $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}$ *consists of* d *eigenvalues in descending order and the corresponding eigenvectors are denoted as the orthogonal matrix* $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d] \in \mathbb{R}^{d\times d}$*. For any nonzero sample* $\boldsymbol{x} \in \mathbb{R}^d$*,* $\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x}$ *will achieve a maximum based on the first* k *eigenvalues if* $\boldsymbol{x}$ *is in the direction of the first* k *eigenvectors of* $\boldsymbol{\Sigma}$*.*

Proof. Given the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$, where $\boldsymbol{\Lambda}_{ii} = \lambda_i$ $(\lambda_1 > \cdots > \lambda_d \ge 0)$, $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_d]$ and $\boldsymbol{V}^\top = \boldsymbol{V}^{-1}$. To simplify the expression, we suppose that every eigenvector satisfies $\|\boldsymbol{v}_i\| = 1$. For any non-zero vector $\boldsymbol{x}$, it can be projected into the space spanned by a set of unit orthogonal bases $\boldsymbol{v}_i \in \mathbb{R}^d$ $(i=1, \cdots, d)$ and its projection $\hat{\boldsymbol{x}}$ can be represented as $\hat{\boldsymbol{x}} = \sum_{i=1}^{d} \alpha_i \boldsymbol{v}_i = \boldsymbol{V}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^d$, and $\boldsymbol{x} = \hat{\boldsymbol{x}} + \Delta\boldsymbol{x}$. Note that, as $\Delta\boldsymbol{x}$ is perpendicular to the space spanned by the bases $\boldsymbol{v}_i|_{i=1}^{d}$, $\Delta\boldsymbol{x}^\top\boldsymbol{\Sigma}$ will always be zero. To make the proof simple and clean, we do not distinguish between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$.

Based on the above conditions, there are:

$$\begin{aligned} &\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} = \boldsymbol{\alpha}^\top\boldsymbol{V}^\top\boldsymbol{\Sigma}\boldsymbol{V}\boldsymbol{\alpha} = \boldsymbol{\alpha}^\top\boldsymbol{\Lambda}\boldsymbol{\alpha} = \sum_{i=1}^{d}\lambda_i\alpha_i^2 \\ &\boldsymbol{x}^\top\boldsymbol{V} = \boldsymbol{\alpha}^\top\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{\alpha}^\top = [\alpha_1, \ldots, \alpha_d] \\ &\boldsymbol{x}^T\boldsymbol{v}_i = \|\boldsymbol{x}\|\|\boldsymbol{v}_i\|\cos\theta_i = \alpha_i, \forall \boldsymbol{v}_i\,, \end{aligned} \tag{4}$$

where θ_i is the angle between $\boldsymbol{x}$ and $\boldsymbol{v}_i$.

First, if $\boldsymbol{x}$ and $\boldsymbol{v}_i$ are collinear in the same direction (*i.e.*, $\theta = 0°$), the maximum of α_i is $\|\boldsymbol{x}\|\|\boldsymbol{v}_i\|$. At this time, $\boldsymbol{x}$ is orthogonal to all the other eigenvectors, *i.e.*, $\boldsymbol{x}^\top\boldsymbol{v}_j = \alpha_j = 0$ $(j \neq i)$, thus $\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} = \lambda_i\alpha_i^2 = \lambda_i\|\boldsymbol{x}\|^2\|\boldsymbol{v}_i\|^2 = \lambda_i\|\boldsymbol{x}\|^2$. Considering the largest eigenvalue λ_1, $\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x}$ can achieve the maximum $\lambda_1\|\boldsymbol{x}\|^2$ in the direction of the first eigenvector $\boldsymbol{v}_1$.

Second, if $\boldsymbol{x}$ is in the direction of both $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ eigenvectors, then $\boldsymbol{x} = \alpha_i\boldsymbol{v}_i + \alpha_j\boldsymbol{v}_j$ and $\boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x} = \lambda_i\alpha_i^2 + \lambda_j\alpha_j^2 = (\lambda_i\cos^2\theta_i + \lambda_j\cos^2\theta_j)\|\boldsymbol{x}\|^2$, where θ_r denotes the angle between $\boldsymbol{x}$ and $\boldsymbol{v}_r$ $(r = i, j)$, respectively. Considering the first two largest eigenvalues λ_1 and λ_2, $\boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x}$ can achieve the maximum $(\lambda_1\cos^2\theta_1 + \lambda_2\cos^2\theta_2)\|\boldsymbol{x}\|^2$ when $\boldsymbol{x}$ is in the direction of first two eigenvectors.

Finally, generalizing to a general case, if $\boldsymbol{x}$ is in the direction of k eigenvectors, considering the first k largest eigenvalues $\lambda_1 > \ldots > \lambda_k$, then $\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x}$ can achieve the maximum $(\lambda_1\cos^2\theta_1 + \cdots + \lambda_k\cos^2\theta_k)\|\boldsymbol{x}\|^2$, when $\boldsymbol{x}$ is in the direction of first k eigenvectors. The proof is completed. □

Here, we also use several local deep descriptors (M d-dimension local deep descriptors) to represent a query image $\boldsymbol{X} \in \mathbb{R}^{d\times M}$ from the query set $\boldsymbol{Q}$. Following the Eq. (3), the corresponding local covariance metric similarities between $\boldsymbol{X}$ and $\boldsymbol{\Sigma}_c^{local}$ can be formalized as below,

$$\boldsymbol{z} = \mathrm{diag}\, f(\boldsymbol{X}, \boldsymbol{\Sigma}_c^{local}) = \mathrm{diag}(\boldsymbol{X}^\top\boldsymbol{\Sigma}_c^{local}\boldsymbol{X})\,, \tag{5}$$

where $\boldsymbol{z} \in \mathbb{R}^M$ contains M local similarities between a query image and one category, $\boldsymbol{\Sigma}_c^{local}$ denotes the local covariance representation calculated by Eq. (2), and $\mathrm{diag}(\cdot)$ returns a column vector of the main diagonal elements of the matrix. Once $\boldsymbol{z}$ is calculated, a global similarity Z can be achieved by linearly weighting M local similarities, *i.e.*, $Z = \boldsymbol{w}^\top\boldsymbol{z}$, where $\boldsymbol{w}$ is the weight vector.

3.3 Model Architecture

The architecture of the proposed CovaMNet shown in Figure 1 mainly contains two modules: a convolutional embedding module and a covariance metric module. Following the previous work (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Yang et al. 2018), the embedding module is composed of four convolutional blocks, where each convolutional block consists of a convolutional layer (with 64 filters of size 3 × 3), a batch normalization layer and a Leaky ReLU layer (instead of ReLU layer). In addition, an additional 2 × 2 max-pooling layer is appended to the first two convolutional blocks, respectively.

Inputting a query image into the embedding module, the output is a $h \times w \times d$ tensor that has M cells ($M = hw$), and each cell denotes a d-dimensional local deep descriptor. Feeding the support set $\boldsymbol{S} = \{\boldsymbol{D}_c\}|_{c=1}^{C}$ into this module, the local covariance representation of the c-th category $\boldsymbol{\Sigma}_c^{local}$ is calculated based on $\boldsymbol{D}_c$ according to Eq. (2), where each subset $\boldsymbol{D}_c$ contains K images for the c-th category.

Subsequently, in the covariance metric module, it calculates the local covariance metric similarities $\boldsymbol{z}$ in Eq. (5) which measures the relation between a query image $\boldsymbol{X}$ and one category $\boldsymbol{\Sigma}_c^{local}$, respectively from the query set $\boldsymbol{Q}$ and the support set $\boldsymbol{S}$. Furthermore, a fully connected (FC) layer is employed to map $\boldsymbol{z}$ into a global similarity Z. Similarly, to measure the relations between this query image and other $C-1$ categories from the support set $\boldsymbol{S}$, other $C-1$ global similarities are obtained in the same way. In this work, all the local similarities between the query image and C categories are concatenated in sequence, then we adopt a 1D convolution layer with a stride of M to realize this process. Finally, a softmax layer with the cross-entropy loss is utilized to get the final classification result.

4 Experiments

In this section, we perform extensive experiments on one common few-shot classification dataset, *i.e., mini*ImageNet, and three fine-grained benchmark datasets, *i.e., Stanford Dogs*, *Stanford Cars* and *CUB Birds*, to evaluate the proposed CovaMNet.

4.1 *mini*ImageNet Few-shot Classification

Dataset The *mini*ImageNet dataset was originally proposed by (Vinyals et al. 2016), a mini-version of ImageNet derived from the ILSVRC-12 dataset (Russakovsky et al. 2015). There are 100 categories with 600 images per category in this dataset and the image resolution is 84 × 84. In this work, we follow the splits of this dataset used in (Ravi and Larochelle 2017), where 64, 16 and 20 categories are for training (auxiliary), validation and testing, respectively.

Experimental Setting Typically, the 5-way 1-shot and the 5-way 5-shot classification tasks are conducted on this dataset. During the process of training, we employ the episodic training mechanism to learn the proposed CovaMNet model. There are totally $300,000$ episodes, each of which is constructed by a support set and a query set. For the 5-way 1-shot classification, there are 5 categories with 15 query images and 1 support image per category, *i.e.,* $5\times15+5\times1 = 80$ images in each episode. Similarly, for the 5-way 5-shot classification, there are $5\times15+5\times5 = 100$ images in each episode. Besides, we adopt Adam algorithm (Kingma and Ba 2015) with an initial learning rate of 5×10^{-3} to optimize our CovaMNet model, where the learning rate is reduced by half for every $100,000$ episodes. During the testing process, 600 episodes are randomly constructed from the

Table 1: The 5-way 1-shot and 5-shot classification accuracies on the *mini*ImageNet dataset, with 95% confidence intervals. The second column refers to which embedded module (Embed. for short) is employed, 32F or 64F. The third column denotes the type of this method, where Metric means this method belongs to the metric-learning based method and Meta belongs to the meta-learning based method. The fourth column indicates whether this method needs to be fine-tuned. * means the result reported by the original work. ‡ denotes that the result is obtained by 20-way training setting.

Model	**Embed.**	**Type**	**Fine Tune**	**5-Way Accuracy (%)**	
				1-shot	5-shot
Baseline k-NN	64F	Metric	N	27.23±1.41	49.29±1.56
Meta-Learner* (Ravi and Larochelle 2017)	32F	Meta	N	43.44±0.77	60.60±0.71
MAML* (Finn, Abbeel, and Levine 2017)	32F	Meta	Y	48.70±1.84	63.11±0.92
SNAIL* (Mishra et al. 2018)	32F	Meta	N	45.10±0.00	55.20±0.00
Matching Nets FCE* (Vinyals et al. 2016)	64F	Metric & Meta	N	43.56±0.84	55.31±0.73
GNN (Garcia and Bruna 2018)	64F	Metric	N	49.02±0.98	63.50±0.84
Prototypical Nets* (Snell, Swersky, and Zemel 2017)	64F	Metric	N	‡49.42±0.78	‡68.20±0.66
Relation Net* (Yang et al. 2018)	64F	Metric	N	50.44±0.82	65.32±0.70
Our CovaMNet	64F	Metric	N	**51.19±0.76**	**67.65±0.63**

testing set to calculate the top-1 mean accuracy as well as the corresponding confidence interval. Note that our proposed CovaMNet model is trained in an end-to-end manner from scratch without requiring the fine-tuning during the testing.

To evaluate the proposed CovaMNet on the *mini*ImageNet dataset, we make comparisons with a baseline model and seven state-of-the-art few-shot learning models, including Baseline k-NN, Matching Nets FCE (Vinyals et al. 2016), Meta-Learner LSTM (Meta-Learner for short) (Ravi and Larochelle 2017), Model-agnostic Meta-learning (MAML) (Finn, Abbeel, and Levine 2017), Prototypical Nets (Snell, Swersky, and Zemel 2017), Relation Net (Yang et al. 2018), Graph Neural Networks (GNN) (Garcia and Bruna 2018) and Simple Neural AttentIve Learner (SNAIL) (Mishra et al. 2018). To compare with the baseline method, we utilize the basic convolutional embedding network followed by three fully connected layers (with a dimensionality of 256) to train a 64-class classification network. Once trained, using this network to extract feature representations and employs a k-NN classifier to get the final classification results during the testing. For other compared models, their experimental settings and results are followed by their original work. However, these models use several different network architectures for the embedding modules. For instance, Meta-learner LSTM and MAML employ a four-convolutional network with 32 filters per convolutional layer (32F for short) to reduce overfitting (Finn, Abbeel, and Levine 2017). The metric-learning based methods (*e.g.,* Prototypical Nets) usually employ the same four-convolutional network but with 64 filters per convolutional layer (64F for short). For the sake of fairness, our CovaMNet adopts the 64F version for the embedding module, since CovaMNet belongs to the metric-learning based model. The GNN is re-run by replacing the embedding module with 64F. The results of SNAIL with shallow embedding module (*i.e.,* 32F) are picked from its ablations.

Experimental Results We report the experimental results in Table 1. The second column refers to which embedded module (Embed. for short) is employed, 32F or 64F. The third column denotes the type of this method, where Metric means this method belongs to the metric-learning based method and Meta means this method belongs to a meta-learning based method. The fourth column indicates whether this method needs to be fine-tuned. The last two columns are the 5-way 1-shot and the 5-way 5-shot classification accuracies on the *mini*ImageNet dataset, with 95% confidence intervals. According to Table 1, it can be seen that the classification results of the Baseline k-NN is not as good as other compared models. This is because it does not adopt the episodic training mechanism but just uses the auxiliary set to learn a pre-trained network for the subsequent few-shot classification task, which verifies the significance and effectiveness of the episodic training mechanism.

Here we divide seven state-of-the-art models into two groups and compare them with our proposed CovaMNet, respectively. The first group contains three Meta-learning based methods, *i.e.,* Meta-Learner LSTM, MAML and SNAIL. Compared with these three Meta-learning based methods, our CovaMNet can achieve more competitive results because of using a more informative covariance representation and a more discriminative covariance metric, in a fairly simpler architecture. For example, in the 5-way 1-shot setting, CovaMNet obtains 7.75%, 2.49% and 6.09% improvements over Meta-Learner, MAML and SNAIL, respectively.

Next, we compared CovaMNet with the second group methods, including four Metric-learning based methods: Matching Nets FCE, GNN, Prototypical Nets and Relation Net. It can be seen that CovaMNet gains 7.63%, 2.17%, 1.77% and 0.75% improvements over above four methods, respectively, in the 5-way 1-shot setting. This is owing to the second-order concept representation in our CovaMNet, which is more informative and effective than the first-order concept representation in other methods, *e.g.*, mean repre-

Table 2: 5-way 1-shot and 5-shot classification accuracies on three fine-grained datasets, *i.e., Stanford Dogs, Stanford Cars and CUB Birds*, with 95% confidence intervals.

Model	**Embed.**	**5-Way Accuracy (%)**					
		Stanford Dogs		***Stanford Cars***		***CUB Birds***	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline k-NN	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
Matching Nets FCE	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	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	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 CovaMNet	64F	**49.10±0.76**	**63.04±0.65**	**56.65±0.86**	**71.33±0.62**	**52.42±0.76**	**63.76±0.64**

sentation used in both Prototypical Nets and Relation Net, and graph representation in GNN. In addition, the specifically designed covariance metric is more suitable than a fixed metric in Prototypical Nets or a simple concatenation in Relation Net. In the 5-way 5-shot setting, our model gains 12.34%, 4.15% and 2.33% improvements over the other three methods respectively, except Prototypical Nets. This is because Prototypical Nets was trained on 20-way 15 queries per training episode which is of higher computational complexity and needs more queries, in the 5-way 5-shot setting. When trained with 5-way 15 query per training episode, Prototypical Nets will get a lower accuracy of $66.53 \pm 0.51\%$, which is worse than ours.

4.2 Fine-grained Few-shot Classification

Dataset Three fine-grained benchmark datasets, *i.e., Stanford Dogs*, *Stanford Cars* and *CUB Birds*, are picked to conduct the fine-grained few-shot classification (FGFS) task. There are 120 categories with a total of $20,580$ images in the Stanford Dogs dataset, where 70, 20 and 30 categories are used for training (auxiliary), validation and testing, respectively. The Stanford Cars dataset contains $16,185$ car images of 196 categories, in which 130, 17 and 49 categories are split for training (auxiliary), validation and testing. As for the CUB Birds dataset, it consists of 6033 bird images of 200 species and is split into 130, 20 and 50 accordingly.

Experimental Settings All settings are the same as those of *mini*ImageNet few-shot classification. That is to say, both the 5-way 1-shot and the 5-way 5-shot experiments are conducted on these three datasets and each image is resized to 84×84 for all models.

Different from the generic (GE) image classification task, the fine-grained (FG) image classification task is more challenging due to the less inter-class variation and larger intra-class variation of the fine-grained datasets. Further, considering a more common and natural problem that only very few examples are available for the new categories, *e.g.,* rare flower species or a bird picture under a glimpse, thus the fine-grained few-shot (FGFS) image classification is more difficult than GE and FG tasks. Due to the lack of previous work about FGFS, existing few-shot learning models are barely performed on the aforementioned fine-grained datasets.

To evaluate the performance of CovaMNet on three fine-grained benchmark datasets, a baseline model and three typical few-shot learning models, including Baseline k-NN, Matching Nets FCE (Vinyals et al. 2016), Prototypical Nets (Snell, Swersky, and Zemel 2017) and GNN (Garcia and Bruna 2018), are implemented on these datasets to make comparisons. The experimental setting of Baseline k-NN is the same as that of *mini*ImageNet few-shot classification. Other three compared models are implemented by following their released codes.

Experimental Results In Table 2, it is obvious that the proposed CovaMNet achieves the best performance on three datasets compared with other models. For example, in the 5-way 5-shot setting, CovaMNet gets 19.9%, 15.54%, 14.85% and 0.77% improvements over the above four methods on the Stanford Dogs dataset, since CovaMNet can not only capture the subtle image cues FG relied on, but also utilizes distribution discrepancy to distinguish different categories, which is more suitable than the existing distance metric for FG task.

5 Conclusions

In this paper, we propose a compact and effective model CovaMNet for few-shot learning. The proposed CovaMNet exploits a local covariance representation to represent the underlying distribution for each category and embeds a novel covariance metric into the network to measure the relations between the query images and categories. Furthermore, the theoretical analysis of the distribution consistency is provided to support our motivation. We evaluate CovaMNet on both generic few-shot classification benchmark dataset and more challenging fine-grained few-shot benchmarks, and achieve competitive results compared with several state-of-the-art models.

Acknowledgements

This work is supported by the National NSF of China (Nos. 61432008, 61806092, U1435214), Jiangsu Natural Science Foundation (No. BK20180326), and Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (No. CX201814).

References

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

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

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T. 2016. Compact bilinear pooling. In *CVPR*, 317–326.

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

Harandi, M. T.; Salzmann, M.; and Porikli, F. M. 2014. Bregman divergences for infinite dimensional covariance matrices. In *CVPR*, 1003–1010.

Huang, Z.; Wang, R.; Shan, S.; Li, X.; and Chen, X. 2015. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 720–729.

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

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

Li, P.; Xie, J.; Wang, Q.; and Zuo, W. 2017. Is second-order information helpful for large-scale visual recognition. In *ICCV*, 2070–2078.

Li, P.; Xie, J.; Wang, Q.; and Gao, Z. 2018. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *CVPR*, 130–138.

Lin, T.-Y., and Maji, S. 2017. Improved bilinear pooling with cnns. In *BMVC*.

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2015. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 1449–1457.

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2018. Bilinear convolutional neural networks for fine-grained visual recognition. *TPAMI* 40(6):1309–1322.

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

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

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

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

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

Tabia, H.; Laga, H.; Picard, D.; and Gosselin, P. H. 2014. Covariance descriptors for 3d shape matching and retrieval. In *CVPR*, 4185–4192.

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

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

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

Tuzel, O.; Porikli, F.; and Meer, P. 2006. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 589–600.

Tuzel, O.; Porikli, F.; and Meer, P. 2007. Human detection via classification on riemannian manifolds. In *CVPR*.

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

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

Wang, R.; Guo, H.; Davis, L. S.; and Dai, Q. 2012. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, 2496–2503.

Wang, L.; Zhang, J.; Zhou, L.; Tang, C.; and Li, W. 2015a. Beyond covariance: Feature representation with nonlinear kernel matrices. In *ICCV*, 4570–4578.

Wang, W.; Wang, R.; Huang, Z.; Shan, S.; and Chen, X. 2015b. Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets. In *CVPR*, 2048–2057.

Wang, W.; Wang, R.; Shan, S.; and Chen, X. 2017. Discriminative covariance oriented representation learning for face recognition with image sets. In *CVPR*, 5749–5758.

Wang, Q.; Li, P.; and Zhang, L. 2017. G2denet: Global gaussian distribution embedding network and its application to visual recognition. In *CVPR*, volume 1, 3.

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