Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples*

Xiu-Shen Wei[1,2] Peng Wang[3] Lingqiao Liu[3] Chunhua Shen[3] Jianxin Wu[1]
[1]Nanjing University [2]Megvii Inc. (Face++) [3]University of Adelaide
{weixs, wujx}@lamda.nju.edu.cn, {peng.wang, lingqiao.liu, chunhua.shen}@adelaide.edu.au

Abstract

Humans are capable of learning a new fine-grained concept with very little supervision, e.g.*, few exemplary images for a species of bird, yet our best deep learning systems need hundreds or thousands of labeled examples. In this paper, we try to reduce this gap by studying the fine-grained image recognition problem in a challenging few-shot learning setting, termed few-shot fine-grained recognition (FSFG). The task of FSFG requires the learning systems to build classifiers for novel fine-grained categories from few examples (only one or less than five). To solve this problem, we propose an end-to-end trainable deep network which is inspired by the state-of-the-art fine-grained recognition model and is tailored for the FSFG task.*

Specifically, our network consists of a bilinear feature learning module and a classifier mapping module: while the former encodes the discriminative information of an exemplar image into a feature vector, the latter maps the intermediate feature into the decision boundary of the novel category. The key novelty of our model is a "piecewise mappings" function in the classifier mapping module, which generates the decision boundary via learning a set of more attainable sub-classifiers in a more parameter-economic way. We learn the exemplar-to-classifier mapping based on an auxiliary dataset in a meta-learning fashion, which is expected to be able to generalize to novel categories. By conducting comprehensive experiments on three fine-grained datasets, we demonstrate that the proposed method achieves superior performance over the competing baselines.

1. Introduction

Fine-grained image recognition, as an important computer vision problem, has attracted tremendous attention and observed rapid performance boost thanks to the sophisticated deep network structures. However, the large-scale fine-grained data volume required to train such classification algorithms limits the ranges where they can be successfully applied to, *e.g.*, very sparse training samples can be collected for some rare bird species. Humans, in contrast, are capable of learning a new fine-grained concept with very little supervision. To mimic this human ability, in this work, we study the fine-grained image recognition in a more practical and challenging few-shot setting, that is, we aim to learn the classifiers of novel fine-grained categories from very few labeled training examples (*a.k.a.* exemplars, usually 1 or 5).

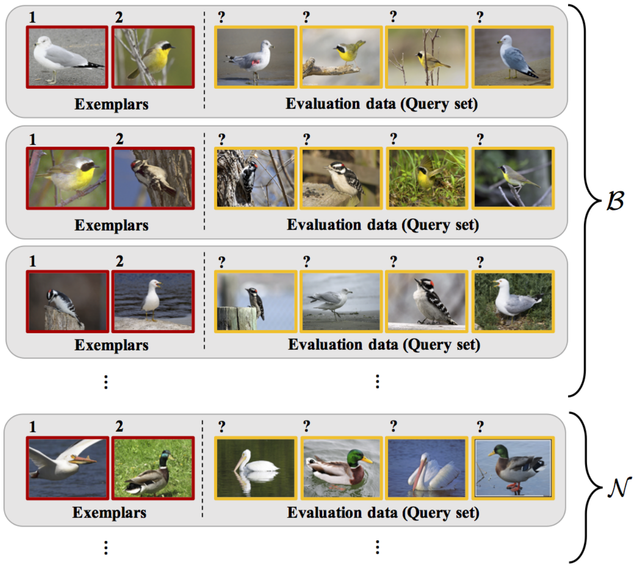


Figure 1: Illustration of the few-shot fine-grained image recognition (FSFG) task. The aim is to learn the classifier for a fine-grained category, bird species in this example, from few exemplars. We train the exemplar-to-classifier mapping based on an auxiliary dataset $\mathcal{B}$ and test the FSFG performance on another dataset $\mathcal{N}$. There are no category overlaps between these two sets.

Learning a classifier for a fine-grained category identified by few exemplars is a challenging problem, as satisfactory classification performance can be expected only when the

*Part of the work was done when X.-S. Wei was visiting The University of Adelaide. The first two authors contributed equally to this work.

learned classifiers can capture the subtle differences between categories and is able to generalize beyond the very limited supervisions. To realize such exemplar-to-classifier mapping, we propose an end-to-end trainable network which is inspired by state-of-the-art fine-grained recognition model and is tailored for the FSFG task. Specifically, the network consists of a bilinear feature learning module and a classifier mapping module. While the former encodes the discriminative information of exemplar image into a feature vector, the latter, as the key part of the network, maps the intermediate image features into the category-level decision boundaries. Two problems remain to succeed with such mappings. On one hand, the distribution of the image-level representation can be complex which poses a great challenge for the mapping. On the other hand, the feature generated from bilinear pooling is very high dimensional, which further impedes the mapping due to the risk of parameter explosion.

The key novelty of our model to mitigate these problems is a "piecewise mappings" function in the classifier mapping module, which generates the decision boundary via learning a set of more attainable sub-classifiers in a much more parameter-economic way. Due to the outer product computation in bilinear pooling, the feature obtained, by nature, can be viewed as a set of sub-vectors, each of which implicitly attends to part of the image. We perform the sub-vector to sub-classifier mapping resorting to highly non-linear mappings. Then, these sub-classifiers are recombined into a global classifier so that it can tell samples from different categories. Intuitively, we learn the feature-to-classifier mapping based on the implicit "part" which may encode simpler and purer information and consequently makes the mapping easier. As a by-product, the piecewise mappings significantly reduce the number of model parameters and enable a more efficient computation. We learn the exemplar-to-classifier mapping using an auxiliary dataset in a meta-learning fashion as shown in Fig. 1. The aim in the meta-training phase is to learn a "mapping paradigm" which is expected to be able to generalize to novel categories.

In experiments, we perform the proposed FSFG method on three fine-grained benchmark datasets, *i.e.*, *CUB Birds* [23], *Stanford Dogs* [9], *Stanford Cars* [11]. Empirical results show that our FSFG model significantly outperforms competing baseline methods.

In summary, our major contributions are three-fold:

- We study fine-grained image recognition in a challenging few-shot setting and propose a novel meta-learning strategy to address this problem.
- We devise a novel exemplar-to-classifier mapping strategy, named piecewise mappings, which resorts to the special structure of the bilinear CNN features to learn a discriminative classifier in a parameter-economic way.
- We conduct comprehensive experiments on three fine-grained benchmark datasets, and our proposed model achieves superior performance over competing solutions on all these datasets.

2. Related work

As our work is related to both fine-grained image recognition and generic few-shot learning, in this section we will briefly review these two topics separately.

2.1. Fine-grained image recognition

Fine-grained recognition is a challenging problem and has recently emerged as an active topic [9, 11, 23]. Over the past decade, fine-grained recognition has achieved high performance levels thanks to the integration of powerful deep learning techniques with large annotated training datasets. A number of effective fine-grained recognition methods have been developed in the literature [2, 5, 7, 8, 16, 17, 28]. Among them, some work, *e.g.*, [8, 17], attempted to learn a more discriminative feature representation by developing powerful deep models. Some methods aligned the objects in fine-grained images to eliminate pose variations and the influence of camera position, *e.g.*, [2, 16]. Moreover, some of them relied on localizing discriminative parts with/without strong supervisions, *e.g.*, [5, 7, 16].

However, current fine-grained recognition systems assume a set of categories known *a priori*, despite the obviously dynamic and open nature of the visual world [1, 26, 25]. Compared with previous work, we are the first to study fine-grained image recognition in a challenging few-shot learning setting where the model is required to recognize novel fine-grained categories by only a few labeled images.

2.2. Generic few-shot image recognition

Nowadays, few-shot image recognition (*a.k.a.* few-shot learning or low-shot learning) [1, 22] has attracted more and more attentions in computer vision and pattern recognition. This line of research explores the possibility of endowing learning systems the ability of rapid learning for novel categories from a few examples. More specifically, these systems are able to learn new concepts on the fly, from few or even a single example as in one-shot learning. Few-shot image recognition is usually tackled by using generative models [15, 19] or, in a discriminative setting, using ad-hoc solutions such as exemplar support vector machines [18]. While recently, many methods solved it in a learning-to-learn formulation [4, 24, 25, 26, 27].

However, previous few-shot image recognition studies all focused on generic images (*e.g.*, images of the ImageNet [20] and CIFAR [12] datasets) or generic patterns (*e.g.*, characters of the Omniglot [14] dataset). Compared with those tasks, we consider a novel few-shot image recognition topic, *i.e.*, few-shot fine-grained image recognition. The most different point of our topic from the generic few-shot image recognition is that, fine-grained recognition relies on more subtle image cues which makes it considerably more challenging. We demonstrate that the proposed model, especially our piecewise mappings component, can cater to the desire of capturing the subtle differences in a fine-grained scenario from limited training data, even one-shot.

3. Learning few-shot fine-grained learners

In this section, we firstly present our learning strategy for FSFG and introduce the relevant notations. Then, a detailed elaboration of various aspects of our method will be followed in the subsequent sections.

3.1. Learning strategy and notations

Our work is built upon the framework of meta-learning which treats the classifier generation process as a mapping function from the few labeled training samples of a category, called "exemplars" hereafter, to their corresponding category classifier. Fig. 2 shows the key idea of this learning scheme. This *exemplar-to-classifier* mapping is learned on an auxiliary training set $\mathcal{B}$. It contains N labeled training images $\mathcal{B} = \{(\mathcal{I}_1, y_1), (\mathcal{I}_2, y_2), \dots, (\mathcal{I}_N, y_N)\}$, where $\mathcal{I}_i$ is an example image and $y_i \in \{1, 2, \dots, C_\mathcal{B}\}$ is its corresponding label. Once the mapping function is learned, it will be applied on another testing set $\mathcal{N}$ to evaluate its performance, where $\mathcal{N}$ contains images of novel categories that do not appear in $\mathcal{B}$.

To train the mapping function, we randomly sample a set of "meta-training sets" from $\mathcal{B}$. Each meta-training set (corresponding to a training episode) contains $C_\mathcal{E} < C_\mathcal{B}$ randomly chosen categories and a few images associated with them. A meta-training set is composed of an "exemplar set" $\mathcal{E}$ and a "query set" $\mathcal{Q}$ to mimic the scenario at the testing stage. Specifically, $\mathcal{E}$ contains N_e (*e.g.*, 1 or 5) exemplar images per category. The query set $\mathcal{Q}$ is coupled with $\mathcal{E}$ (has the same categories), but has no overlapped images. Each category of $\mathcal{Q}$ contains N_q query images. During training, $\mathcal{E}$ will be fed into the to-be-learned mapping function M to generate the category classifiers $F_\mathcal{E}$:

$$\mathcal{E} \xrightarrow{M} F_\mathcal{E} \, . \tag{1}$$

Then, $F_\mathcal{E}$ are subsequently applied to $\mathcal{Q}$ for evaluating the classification loss. The training objective then amounts to learning the mapping function by minimizing the classification loss. This process is formally written as follows:

$$\min_{\lambda} \underset{\{\mathcal{E},\mathcal{Q}\}\sim\mathcal{B}}{E} \left\{ \mathcal{L}\left(F_\mathcal{E} \circ \mathcal{Q}\right) \right\} , \tag{2}$$

where λ denotes the model parameters of the mapping function M (from $\mathcal{E}$ to $F_\mathcal{E}$), and $\mathcal{L}$ is the loss function. $F_\mathcal{E} \circ \mathcal{Q}$ denotes applying the category classifiers $F_\mathcal{E}$ generated by the exemplar set $\mathcal{E}$ on the query set $\mathcal{Q}$.

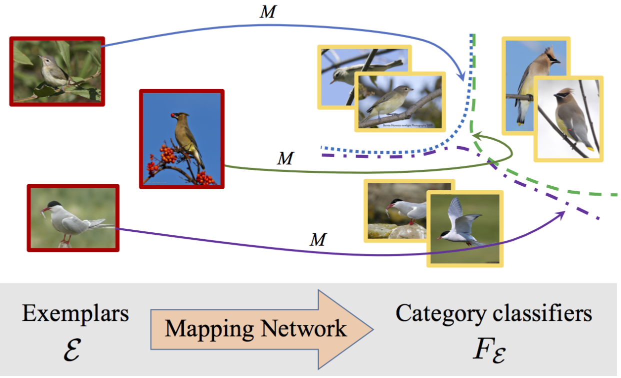


Figure 2: Key idea of the proposed FSFG model. In each episode, we sample an exemplar set $\mathcal{E}$ from $\mathcal{B}$, which is composed of a subset of categories (three categories in this example) and each category contains few exemplars (the images with red border). We wish to learn a mapping M that can map these exemplars into their corresponding category classifiers (the dashed lines). The mapping parameters are learned so that these classifiers can correctly distinguish the query images (the images with yellow border).

3.2. Model

We implement the above exemplar-to-classifier mapping by adopting a trainable neural network. Fig. 3 shows the overall architecture of the network. As we can see, the network is composed of two modules: a representation learning module and a classifier mapping module. While the former adopts a bilinear CNN structure to encode the discriminative information of an exemplar image into a high-dimensional feature vector, the latter, as the key part of the network, maps the intermediate image representation into a category classifier. In the next two sub-sections, we elaborate these two modules in more details.

3.2.1 Representation learning

We employ a bilinear CNN (BCNN) structure [17] to learn the image representation considering its state-of-the-art performance in fine-grained image recognition. BCNN consists of two feature extractors whose outputs are multiplied using outer product at each location of the image and pooled to obtain an image representation. Concretely, given two convolutional networks (A and B) as two streams of BCNN, we assume their outputs are re-organized into $f_A(\mathcal{I}) \in \mathbb{R}^{n_A \times L}$ and $f_B(\mathcal{I}) \in \mathbb{R}^{n_B \times L}$, where n_A, n_B denotes the dimensionality of the outputs and L denotes the spatial locations. Then, at location l, the bilinear representation will be $\mathbf{b}_l \in \mathbb{R}^{n_A \times n_B}$,

$$\mathbf{b}_l = f_A(l, \mathcal{I}) f_B(l, \mathcal{I})^\intercal \, . \tag{3}$$

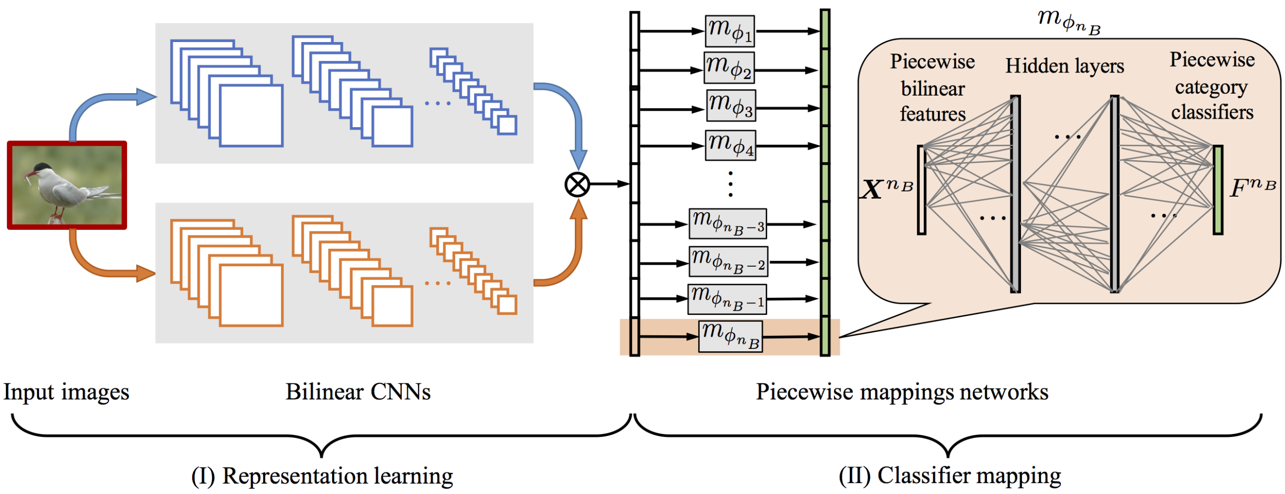


Figure 3: Overview structure of our proposed FSFG model. On the left, it is the first component (the bilinear pooling module) for representation learning. On the right, the second component (the classifier mapping module) mapps the intermediate image features into the category classifiers.

The vectorized versions of $\{\mathbf{b}_l\}$ will be pooled over the entire image to derive the image representation $\boldsymbol{x} \in \mathbb{R}^{D\times 1}$ (for interpretation simplicity we let $D = n_A \times n_B$), that is,

$$\boldsymbol{x}(\mathcal{I}) = \sum_{l=1}^{L} \text{vec}(\mathbf{b}_l)\,. \tag{4}$$

With the outer product computation, bilinear structure modulates one feature stream with another. Thus, the BCNN feature $\boldsymbol{x}$ can be viewed as a set of n_B sub-vectors $\boldsymbol{x}^t$:

$$\boldsymbol{x} = \left[\boldsymbol{x}^1; \boldsymbol{x}^2; \ldots; \boldsymbol{x}^t; \ldots; \boldsymbol{x}^{n_B}\right], \forall t : \boldsymbol{x}^t \in \mathbb{R}^{n_A\times 1}\,, \tag{5}$$

where $\boldsymbol{x}^t$ is the modulated feature of f_A by the t-th feature of f_B. This is similar to the multiplicative feature interactions in attention mechanisms [17]. From the observation that each modulated feature map tends to focus on an implicit "part" of an object, and thus, $\boldsymbol{x}^t$ can be viewed as the feature description for that "part". In our implementation, we train the bilinear CNN by performing the same procedure in [17] and use it as the image representation extractor.

To represent a set of N_e exemplar images belonging to category k, we simply compute the mean image representation as the category-level representation by:

$$\boldsymbol{X}_k = \frac{1}{N_e}\sum_{i=1}^{N_e} \boldsymbol{x}_i\,, \tag{6}$$

where $\{\boldsymbol{x}_i\}$ are samples with $y_i = k$.

3.2.2 Classifier mapping

Now that the information of each category identified by few exemplars has been encoded into a bilinear feature vector, the task of the classifier mapping module is to map these intermediate category-level representations into their corresponding category classifiers. Mathematically, this module computes a D-dimensional classifier $F_k \in \mathbb{R}^D$ for each category through a mapping $M : \mathbb{R}^D \to \mathbb{R}^D$.

A straightforward solution to realize this mapping is via a global mapping, either linear or nonlinear. For example, a linear mapping can be:

$$F_k = \mathbf{W}_g \boldsymbol{X}_k + \mathbf{b}_g\,, \tag{7}$$

where $\mathbf{W}_g \in \mathbb{R}^{D\times D}$ and $\mathbf{b}_g \in \mathbb{R}^D$ denote the parameters of the global mapping. However, this mapping strategy suffers from two drawbacks. First, as the feature $\boldsymbol{X}_k$ is supposed to encode the category-level information, the distribution of which can be highly complex. This poses a great challenge for the global mapping to find a decision boundary in such a complex feature space. Second, since the bilinear feature tends to be high dimensional, this mapping may result in parameter explosion, which will make the network training hard or infeasible.

To mitigate these problems, we propose a novel "piecewise mappings" strategy, which exploits the structure of the bilinear features. As analyzed in Sec. 3.2.1, the bilinear feature $\boldsymbol{X}_k$ can be viewed as a set of sub-vectors $\boldsymbol{X}_k^t$ with each sub-vector describes an implicit "part" of the object. Intuitively, we can test if an object falls into the category described in the exemplars by checking whether each "part" of it is compatible with the exemplars. This motivates us to apply a piecewise mapping to first map each sub-vector $\boldsymbol{X}_k^t$ into its corresponding sub-classifier F_k^t, and then combine these sub-classifiers together to generate the global category classifier. Fig. 3 shows this mapping with more details.

Concretely, a sub-vector $\boldsymbol{X}_k^t$ is firstly mapped into a sub-classifier F_k^t via a nonlinear multilayer perceptron (MLP) $m_{\phi_t}(\cdot)$ as

$$F_k^t = m_{\phi_t}(\boldsymbol{X}_k^t)\,. \tag{8}$$

We learn n_B such MLPs $\{m_{\phi_t}(\cdot)\}$ to derive n_B sub-classifiers $\{F_k^t\}$, and then these sub-classifiers are concatenated together to generate the global category classifier F_k:

$$F_k = [F_k^1; F_k^2; \dots; F_k^{n_B}]\,. \tag{9}$$

Essentially, our model simplifies the global mapping approach by assuming that the classifier for the t-th sub-vector is solely determined by the information from the t-th sub-vector in the exemplar set. Despite resulting more restrictive mapping function, this assumption makes the network much easier to train. Note that, this mapping scheme will significantly reduce the model parameters involved in classifier generation. Taking one-layer mapping for example, let's assume $n_A = n_B = 512$. For the global mapping, it requires more than 512^4 parameters. For the proposed piecewise mappings, however, the number is reduced to about 512^3. In addition, although there are parameter-economy variants of BCNN [6], our piecewise classifier mappings still show better performance. This suggests that the proposed classifier mapping function brings benefits more than merely reducing the model size (cf. Table 2).

3.2.3 Network training

Given a query sample $\boldsymbol{x}$ with label $y = c$, we compute its prediction distribution via softmax as:

$$p_M(y = c|\boldsymbol{x}) = \frac{\exp(F_c \cdot \boldsymbol{x})}{\sum_{c'} \exp(F_{c'} \cdot \boldsymbol{x})}\,. \tag{10}$$

The model parameters are trained via minimizing the negative log-likelihood $\mathcal{J}(\boldsymbol{x}, y) = -\log(p_M(c|\boldsymbol{x}))$. With this, we can now summarize the training in an episode as follows. First, we select an exemplar set $\mathcal{E}$ from $\mathcal{B}$ and learn/generate the classifiers $F_{\mathcal{E}}$. Then, we establish a query set $\mathcal{Q}$. The model parameters are optimized by minimizing $\mathcal{J}(\mathcal{Q})$. Algorithm 1 illustrates the training process in more details.

4. Experiments

In this section, we first describe the experimental setups, implementation details and the datasets used in experiments. Then, we present the few-shot fine-grained image recognition results on three fine-grained benchmark datasets. Finally, ablation studies are given to further evaluate the effectiveness of our proposed classifier mapping strategy.

Algorithm 1 Training episode loss computation for the proposed piecewise mappings.

```
Require: $\mathcal{B}$ is an auxiliary training set with $N$ images belonging
    to $C_{\mathcal{B}}$ categories; $\mathcal{B}_c$ denotes a subset of $\mathcal{B}$ containing all images
    belonging to the $c$-th category; $C_{\mathcal{E}}$ denotes the number
    of categories in an exemplar set $\mathcal{E}$ as well as a query set $\mathcal{Q}$ for
    an episode; $\mathcal{E}_k$ denotes the elements $(\boldsymbol{x}_i, y_i = k)$ in $\mathcal{E}$ with
    element size $N_e$; $\mathcal{Q}_k$ denotes the elements $(\boldsymbol{x}_j, y_j = k)$ in $\mathcal{Q}$
    with element size $N_q$; $n$ denotes the number of piecewise mappings;
    RandomSample($\mathcal{T}, N$) denotes a set of $N$ elements
    chosen uniformly at random from set $\mathcal{T}$, without replacement;
    $\mathcal{S}$ denotes a category set and $S_i$ denotes its $i$-th element.
Select a category subset $\mathcal{S}$ for an episode
$\mathcal{S} \leftarrow$ RandomSample($\{1, 2, \dots, C_{\mathcal{B}}\}, C_{\mathcal{E}}$);
for $k$ in $\{1, 2, \dots, C_{\mathcal{E}}\}$ do
    Select $\mathcal{E}_k \leftarrow$ RandomSample($\mathcal{B}_{S_k}, N_e$);
    Compute the category-level representation $\boldsymbol{X}_k$ following
    Eq. 6;
    Generate the category classifier $F_k$ by Eq. 8 and Eq. 9;
    Select $\mathcal{Q}_k \leftarrow$ RandomSample($\mathcal{B}_{S_k} \backslash \mathcal{E}_k, N_q$);
end for
Initialize loss $\mathcal{J} \leftarrow 0$;
for $k$ in $\{1, 2, \dots, C_{\mathcal{E}}\}$ do
    for $(\boldsymbol{x}, y)$ in $\mathcal{Q}_k$ do
        $\mathcal{J} \leftarrow \mathcal{J} + \mathcal{J}(\boldsymbol{x}, y)$;
    end for
end for
$\mathcal{J} = \frac{\mathcal{J}}{C_{\mathcal{E}} \times N_q}$
Update model parameters by minimizing $\mathcal{J}$;
return  $n$ piecewise mappings $[m_{\phi_1}; \dots; m_{\phi_n}]$.
```

Table 1: Category split for three datasets. C_{total} denotes the total number of categories in a dataset, $C_{\mathcal{B}}$ denotes the number of categories in $\mathcal{B}$ and $C_{\mathcal{N}}$ denotes the number of categories in $\mathcal{N}$.

♯ category	*CUB Birds*	*Stanford Dogs*	*Stanford Cars*
C_{total}	200	120	196
$C_{\mathcal{B}}$	150	90	147
$C_{\mathcal{N}}$	50	30	49

4.1. Datasets, setups and implementation details

Our experiments are conducted on three fine-grained benchmark datasets, *i.e.*, *CUB Birds* (200 categories of birds, 11,788 images) [23], *Stanford Dogs* (120 categories of dogs, 20,580 images) [9], *Stanford Cars* (196 categories of cars, 16,185 images) [11]. For each dataset, we randomly split its original image categories into two disjoint subsets: one as the auxiliary training set $\mathcal{B}$, and the other as the FSFG testing set $\mathcal{N}$. Table 1 presents the details of the category split. For each category in $\mathcal{B}$, we follow the raw splits provided by these datasets to split the data into training and validation. While the former is used to train the parameters, the latter is used to monitor the learning process.

To mimic the testing condition, in each training episode, we set the category size of the exemplar set $\mathcal{E}$ to be same as the number of categories in the testing set $\mathcal{N}$, *i.e.*, $C_{\mathcal{E}} = C_{\mathcal{N}}$. Further we set $N_e = 1$ ($N_e = 5$) for one-shot learning (five-shot learning) and N_q is set to be 20 in all settings. Similarly, during the testing phase, for each category in $\mathcal{N}$, we randomly choose one exemplar (five exemplars) for one-shot learning (five-shot learning), and another 20 samples are randomly selected to evaluate the recognition performance. We repeat this evaluation process twenty times, and the mean classification accuracy is used as the evaluation criterion.

In theory, we can choose any network structures as the base network for our bilinear feature learning module. Since our key contribution is in the classifier mapping scheme, we choose AlexNet [13] as the two streams in BCNN, considering the trade off between its representation capacity and computational efficiency. Specifically, we adopt the AlexNet model pre-trained on the Places 205 database [29] to initialize the representation learning parameters. The reason why we use the Place dataset [29] instead of ImageNet [20] is to avoid the FGFS testing categories to be present in the pre-training dataset. We fine-tune the bilinear feature learning module on the auxiliary training set first and freeze it during the classifier learning process. For the classifier mapping module, without otherwise stated, we choose the mapping function m_{ϕ_t} to be a three-layer MLP, where 1024 hidden units are adopted in each layer and Exponential Linear Units (ELU) [3] is used in each layer as the non-linear activation function. SGD is used to optimize the parameters with learning rate of 0.1. We implement our model using the open-source library PyTorch.

4.2. Main results

We present the main results of FSFG by firstly introducing some baseline methods and then reporting the empirical results on these three datasets.

4.2.1 Comparison methods

In our experiments, we compare our proposed model to the following competitive baselines. Note that, apart from the original bilinear CNN, we also implement a compact bilinear CNN [6] as the image feature extractor to facilitate the comparison, which enables much lower feature dimensionality but keeps almost the same classification discriminative ability [6]. For compact bilinear pooling, we follow the optimal settings suggested in [6]. The dimensionality of compact bilinear pooling representations is $8,192$-d (much less than $65,536$-d of fully bilinear pooling). In our empirical results, the results of compact bilinear pooling are denoted as "CB" in Table 2, and the results of fully bilinear pooling are denoted as "FB".

- k**-NN** (k-nearest neighbors): Following the testing setting introduced in Sec. 4.1, we choose one sample (five samples) for each category in $\mathcal{N}$ as exemplar(s) and 20 samples in the same category for evaluation. We use the BCNN (either original or compact version) fine-tuned on $\mathcal{B}$ as the image representation extractor, and nearest neighbor is adopted as the classifier to categorize the evaluation images. Specifically, the image representations are first ℓ_2-normalized and cosine distance is used as the distance metric. Note that, for five-shot learning, the representations of five exemplars are averaged before normalization to serve as the category-level representation. This process will be repeated twenty times as for our method. (This applies to all other baselines, so we omit this when introducing the following baselines.)

- **SVM** (support vector machine): After obtaining the bilinear representations for exemplars of the testing categories in $\mathcal{N}$, we train a classifier for each category based on these representations. In particular, for one-shot learning, this baseline becomes exemplar-SVMs [18].

- **Siamese-Net** [10]: As a standard metric-learning strategy, Siamese-Net is a competitive solution for few-shot learning. It learns a feature space in which images of the same category are close but images belonging to different categories are separated apart. We train a Siamese-Net based on $\mathcal{B}$ by sampling pair-wise examples and the corresponding binary labels ("1" presents examples are from the same category and "0" is not.) Similar to [10], the regularized cross-entropy loss on the binary classifier is used. During evaluation, Siamese-Net could rank similarities between exemplars and testing data.

- **Global mapping**: As aforementioned in Sec. 3.2.2, an alternative solution to our proposed piecewise classifier mappings is global mapping. It follows the idea of the global feature to global classifier mapping by applying the mapping function directly on the category-level representation.

4.2.2 Comparison results

Table 2 presents the average accuracy rates of FSFG on the novel categories of three fine-grained datasets. For each dataset, we report both one-shot and five-shot recognition results. As shown in that table, our proposed model consistently and significantly outperforms the other baseline methods on these datasets.

Generally, we see the simple baseline k-NN performs well and it even outperforms other more sophisticated baselines on some settings, *e.g.*, on *Stanford Dogs*. This is due to the discriminative capacity of the bilinear CNN features. SVM observes more obvious advantage comparing to k-NN when

Table 2: Comparison results (mean±std.) on three fine-grained datasets. The highest average accuracy of each column is marked in bold. "●/○" denotes that our proposed model performs significantly better/worse than the corresponding method by the pairwise t-test with confidence level 0.05. "FB" stands for using the fully bilinear pooling representations, and "CB" is for using compact bilinear pooling.

Method	*CUB Birds*		*Stanford Dogs*		*Stanford Cars*	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
k-NN (FB)	38.85±3.43 ●	55.58±0.84 ●	24.53±2.36 ●	40.30±2.34 ●	26.99±2.91 ●	43.40±1.68 ●
k-NN (CB)	24.52±1.80 ●	41.85±1.51 ●	18.31±1.81 ●	32.37±1.15 ●	21.25±1.78 ●	39.42±1.57 ●
SVM (FB)	34.47±1.93 ●	59.19±1.28 ●	23.37±3.18 ●	39.50±1.07 ●	25.66±1.53 ●	51.07±1.51
SVM (CB)	24.94±1.97 ●	41.93±1.69 ●	18.25±2.83 ●	30.50±1.76 ●	21.34±1.94 ●	39.43±1.46 ●
Siamese-Net (FB)	37.38±1.53 ●	57.73±1.38 ●	23.99±1.66 ●	39.69±1.17 ●	25.81±1.67 ●	48.95±1.31 ●
Siamese-Net (CB)	26.58±2.47 ●	43.51±1.53 ●	19.28±2.60 ●	31.49±1.22 ●	22.41±1.55 ●	40.07±1.88 ●
Global mapping (FB-)	24.12±1.39 ●	34.59±1.77 ●	20.55±1.48 ●	30.93±1.91 ●	20.50±1.60 ●	30.58±1.82 ●
Global mapping (CB)	25.42±2.22 ●	36.37±1.04 ●	20.77±2.75 ●	32.33±2.11 ●	20.24±1.94 ●	32.66±1.86 ●
Ours	**42.10±1.96**	**62.48±1.21**	**28.78±2.33**	**46.92±2.00**	**29.63±2.38**	**52.28±1.46**

exploiting five training exemplars. Siamese-Net, as another discriminative method, achieves comparable performance to SVM but is outperformed by our method. This reflects our meta-learning strategy can better generalize to unseen/novel fine-grained categories. For the global mapping, because BCNN generates image representation of ultra-high dimensionality (*i.e.*, $65,536$ in our case), it is infeasible to learn a global mapping on such high-dimensional feature vectors. In order to realize the global mapping, we apply an additional linear mapping to first reduce $65,536$-d features into $8,192$-d feature vectors, and based on the low-dimensional features, we conduct the global mapping. It is denoted as "Global mapping (FB-)" in Table 2. The global mapping is also implemented as a three-layer networks. As seen, our proposed piecewise mappings significantly outperforms the global mapping. In ablation studies, we will further compare these two types of mapping schemes.

Another interesting observation here is that the few-shot recognition performance gap between FB and CB is large. Note that, both FB and CB are trained on the same training set and achieve comparable classification performance on the validation set. This phenomenon may be explained as that the CB feature is not suitable for similarity matching (*i.e.*, the experimental case of the testing set). It is an open problem worth further explorations.

4.3. Ablation studies

To further inspect our piecewise mappings strategy for FSFG, we conduct ablation experiments on two aspects. First, we compare the global mapping and piecewise mappings on a fairer setting. Second, we investigate the influence of the mapping function m_{ϕ_t} variations on the FSFG performance.

4.3.1 Piecewise mappings *vs.* global mapping

As aforementioned, due to high-dimensionality of bilinear feature, it is infeasible to learn a non-linear (even a simple linear) global mapping on the original bilinear features (*e.g.*, $65,536$ dimensionality) in practice. To perform the global mapping, we modify the original AlexNet structure by reducing the number of units of the last convolution layer from 256 to 64. By doing this, the bilinear feature becomes $64 \times 64 = 4096$-dimensionality, which is feasible to learn a non-linear global mapping. In experiments, a three-layer MLP acts as the global mapping. The hidden units number is selected via cross-validation based on a set of $\{4096, 8192, 16384, 20480\}$. Finally, $16,384$ hidden units are selected because of its optimal performance.

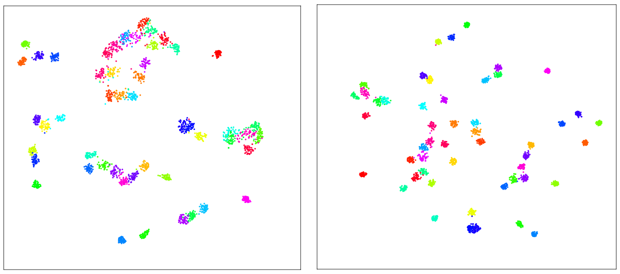
For our proposed piecewise mappings, based on the modified BCNN, the piecewise mappings function is applied to 64-d sub-vectors. Totally, there are 64 piecewise mappings. Each of them is implemented as a three-layer network whose hidden layers contain 256 hidden units. ELU [3] is used as the activation function for both global mapping and piecewise mappings.

Table 3 demonstrates the comparison results of piecewise mappings *vs.* global mapping. Still the piecewise mappings significantly outperform the global mapping on all the three datasets. These observations can serve as a stronger evidence for the superiority of our proposed method.

Apart from the above quantitative evaluation, we present some qualitative results by visualizing the $4,096$-d category classifiers generated by global mapping and piecewise mappings in 2D space in Fig. 4. The dots with the same color denote the classifiers generated from different exemplar images of the same category in $\mathcal{N}$. Different colors represent classifiers of different categories. We randomly select 250 exemplars per category to conduct five-shot recognition. Thus, one category contains 50 versions of classifiers (50 dots in the same one color). As shown in the figure, the classifiers

Table 3: Comparison results of global mapping and piecewise mappings (our proposal) on three datasets. The highest average accuracy of each column is marked in bold. "•" denotes that the piecewise mappings outperform the global mapping with confidence level 0.05 by the pairwise t-test.

Method	*CUB Birds*		*Stanford Dogs*		*Stanford Cars*	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Global mapping	27.36±1.64 •	38.05±1.55 •	19.55±2.27 •	32.53±2.35 •	16.06±2.06 •	26.17±1.02 •
Piecewise mappings (Ours)	**31.00**±**2.85**	**48.80**±**2.33**	**23.07**±**3.24**	**41.02**±**2.50**	**18.98**±**2.18**	**31.51**±**1.38**



(a) By global mapping (b) By our piecewise mappings

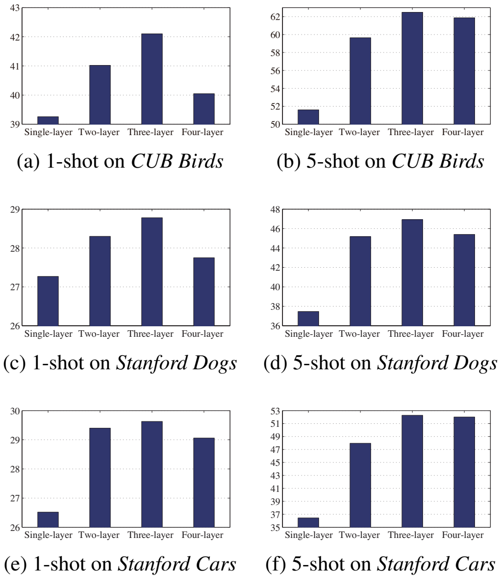
Figure 4: Visualization of the category classifiers generated by global mapping and piecewise mappings in 2D space by t-SNE [21]. Each dot denotes a generated classifier and different colors represent different categories. For each category, fifty classifiers are shown, each of which is obtained via randomly sampled five exemplars. This visualization is based on *CUB Birds*. (The figures are best viewed in color.)

generated by piecewise mappings exhibit better category-separability and more centralized intra-category aggregation. This, in some sense, reflects that the classifiers generated by our method tend to capture the essence of the corresponding categories and maintain better distinguishing capacity.

4.3.2 m_{ϕ_t} with different numbers of layers

We implement the mapping functions m_{ϕ_t} in our classifier mapping module as MLPs. Since the depth plays an important role in determining the modeling capacity of MLPs, in this part, we investigate how the FSFG performance changes *w.r.t.* different number of layers in m_{ϕ_t}. Specifically, we change the number of layers from 1 to 4. The ablation study results are shown in Fig. 5.

Generally, we can see that a single-layer mapping leads to worst performance. This is due to its so limited modeling capacity that cannot realize the complex feature-to-classifier mapping. The FSFG performance rises when adding another layer and peaks when three-layer mappings are used. Beyond that point, continuing to increase the depth of the mapping functions will do harm to the recognition performance, especially in the one-shot scenario. This study necessitates the need to apply a highly non-linear mapping to learn a satisfactory classifier.



(a) 1-shot on *CUB Birds* (b) 5-shot on *CUB Birds*

(c) 1-shot on *Stanford Dogs* (d) 5-shot on *Stanford Dogs*

(e) 1-shot on *Stanford Cars* (f) 5-shot on *Stanford Cars*

Figure 5: Ablation study on m_{ϕ_t} with different number of layers. In each sub-figure, the horizontal axis is the number of layers and the vertical axis represents the accuracy rate.

5. Conclusion

In this paper, we have presented the first study on fine-grained image recognition in a practical and challenging few-shot learning setting, which requires to learn the classifier for a fine-grained category identified by few exemplars. To address this problem, we proposed an end-to-end trainable network which was inspired by the bilinear CNN model and was tailored for the fine-grained few-shot learning task. The key novelty of our network was the piecewise classifiers mapping module. By considering the special structure of bilinear CNN features, it decomposed the exemplar-to-classifier mapping into a set of more attainable "part"-to-"part classifier" mappings. As a by-product, it significantly reduced the model parameters. Through comprehensive experiments on three standard fine-grained image classification dataset, our method showed promising results.

In the future, it appears promising to use transfer learning techniques by leveraging the already gained experience (*e.g.*, the classifiers of the known categories) based on the base set for generalizing the learning ability upon the novel set.

References

[1] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531, Barcelona, Spain, Dec. 2016. 2

[2] S. Branson, G. V. Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. In *British Machine Vision Conference*, pages 1–14, Nottingham, England, Sept. 2014. 2

[3] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proceedings of International Conference on Learning Representations*, pages 1–14, San Juan, Puerto Rico, May. 2016. 6, 7

[4] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of International Conference on Machine Learning*, pages 1–10, Sydney, Australia, Aug. 2017. 2

[5] J. Fu, H. Zheng, and T. Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4438–4446, Honolulu, HI, Jul. 2017. 2

[6] Y. Gao, O. Beijborn, N. Zhang, and T. Darrell. Compact bilinear pooling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326, Las Vegas, NV, Jun. 2016. 5, 6

[7] S. Huang, Z. Xu, D. Tao, and Y. Zhang. Part-stacked CNN for fine-grained visual categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1173–1182, Las Vegas, NV, Jun. 2016. 2

[8] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2008–2016, Montréal, Canada, Dec. 2015. 2

[9] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Fine-Grained Visual Categorization*, pages 806–813, Colorado Springs, CO, Jun. 2011. 2, 5

[10] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of International Conference on Machine Learning*, pages 1–8, New York, NY, Jun. 2016. 6

[11] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. In *Proceedings of IEEE International Conference on Computer Vision Workshop on 3D Representation and Recognition*, pages 554–561, Sydney, Australia, Dec. 2013. 2, 5

[12] A. Krizhevsky and G. E. Hinton. Convolutional deep belief networks on CIFAR-10. *Technique Report*, 2010. 2

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, Lake Tahoe, NV, Dec. 2012. 6

[14] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of Annual Meeting of the Cognitive Science Society*, pages 1–6, Boston, MA, 2011. 2

[15] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[16] D. Lin, X. Shen, C. Lu, and J. Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, Boston, MA, Jun. 2015. 2

[17] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1449–1457, Sandiago, Chile, Dec. 2015. 2, 3, 4

[18] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proceedings of IEEE International Conference on Computer Vision*, pages 89–96, Barcelona, Spain, Nov. 2011. 2, 6

[19] D. J. Rezende, S. Mohamed, I. Danihelka, K. Gregor, and D. Wierstra. One-shot generalization in deep generative models. In *Proceedings of International Conference on Machine Learning*, pages 1521–1529, New York, NY, Jun. 2016. 2

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 6

[21] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 8

[22] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, Barcelona, Spain, Dec. 2016. 2

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. *Techique Report CNS-TR-2011-001*, 2011. 2, 5

[24] Y.-X. Wang and M. Hebert. Learning from small sample sets by combining unsupervised meta-training with CNNs. In *Advances in Neural Information Processing Systems*, pages 244–252, Barcelona, Spain, Dec. 2016. 2

[25] Y.-X. Wang and M. Hebert. Model recommendation: Generating object detectors from few samples. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1619–1628, Boston, MA, Jun. 2015. 2

[26] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *Proceedings of European Conference on Computer Vision*, pages 616–634, Amsterdam, Netherlands, Oct. 2016. 2

[27] S. Yeung, V. Ramanathan, O. Russakovsky, L. Shen, G. Mori, and L. Fei-Fei. Learning to learn from noisy web videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, Honolulu, HI, Jul. 2017. 2

[28] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Transactions on Image Processing*, 25(4):1713–1725, 2016. 2

[29] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, Montréal, Canada, Dec. 2014. 6