Pattern Recognition 122 (2022) 108304

Contents lists available at ScienceDirect

Pattern Recognition

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

Unsupervised descriptor selection based meta-learning networks for few-shot classification

Zhengping Hu[a,b], Zijun Li[a,*], Xueyu Wang[a], Saiyue Zheng[a]

[a] School of Information Science and Engineering, Yanshan University, Qinhuangdao, China
[b] Hebei Key Laboratory of Information Transmission & Signal Processing, Qinhuangdao, China

ARTICLE INFO

Article history:
Received 23 April 2021
Revised 17 July 2021
Accepted 5 September 2021
Available online 10 September 2021

Keywords:
Meta-learning
Few-shot classification
Unsupervised localization
Descriptor selection

ABSTRACT

Meta-learning aims to train a classifier on collections of tasks, such that it can recognize new classes given few samples from each. However, current approaches encounter overfitting and poor generalization since the internal representation learning is obstructed by backgrounds and noises in limited samples. To alleviate those issues, we propose the Unsupervised Descriptor Selection (UDS) to tackle few-shot learning tasks. Specifically, a descriptor selection module is proposed to localize and select semantic meaningful regions in feature maps without supervision. The selected features are then mapped into novel vectors by a task-related aggregation module to enhance internal representations. With a simple network structure, UDS makes adaptation between tasks more efficient, and improves the performance in few-shot learning. Extensive experiments with various backbones are conducted on Caltech-UCSD Bird and *mini*ImageNet, indicate that UDS achieves the comparable performance to state-of-the-art methods, and improves the performance of prior meta-learning methods.

1. Introduction

Recently, deep learning methods have achieved great success in the field of computer vision [1], language processing [2] and medical research [3], *etc*. However, the training of agents requires thousands of labelled samples, which depends upon expensive collection and annotation costs and are usually rare and unavailable in real application. Moreover, the learned experience is not transferable in most of tasks. Different from deep networks, humans always learn transferable experience between different tasks and learns fast only through few information when handling a new task. For example, a child can recognize the zebra in zoo with several corresponding pictures or descriptions although the child has never seen zebra before.

Motivated by the limitation of conventional deep networks and inspired by the adaptation ability of humans in fast learning, few-shot learning [4,5] is increasingly gaining attention. Similar to human intelligence, the few-shot learning method aims to recognize novel visual categories from few labelled samples. When handling a new coming task only given few samples of new class, fast and precise classification is challenging and the agent needs to avoid overfitting. Data augmentation and regularization can alleviate overfitting problem when the prior experience and data are severely limited, but do not completely solve the problem.

In order to learn transferable experience in multi-tasks, Vinyals et al. [6] proposed an effective training mechanism which is called episodes (meta-tasks) in few-shot learning. During the training stage, the model is trained on few labelled samples (Support Set) and tested with queries. Taking 5-way 1-shot classification for example, every episode contains five samples from each class and several query images for test. This meta-learning mechanism mimics few-shot classification between different tasks in reality. The use of episodes makes training stage more efficient and there by improves generalization ability, and have been widely proved in recent few-shot learning methods [7–9]. Based on meta-learning, the current few-shot methods could mainly divided into three groups: metric-learning based methods [6,10], memory based methods [11,12] and optimization based methods [13,14].

The research of metric-learning based model is greatly raising and achieves great success due to its simplicity and prominent performance in few-shot classification. However, state-of-the-art methods tend to deepen backbone and complicate the structure of meta-networks, and have not exceeded human performance in few-shot learning. Quickly capturing new visual contents for fast learning is the hallmark of human intelligence as it can separate related objects from backgrounds and other irrelevant parts. Meta-learning agents should also quickly capture new contents, extract

[*] Corresponding author.
E-mail addresses: hzp@ysu.edu.cn (Z. Hu), lizj@stumail.ysu.edu.cn, 286071970@qq.com (Z. Li), jyq@stumail.ysu.edu.cn (X. Wang), saiyue@stumail.ysu.edu.cn (S. Zheng).

https://doi.org/10.1016/j.patcog.2021.108304
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

sample-related semantic information and obtain internal representations for further meta-learning and testing. Therefore, the descriptor selection module in this paper is proposed to utilize semantic information of the image and localize interesting regions. The descriptor selection module can discard backgrounds without supervision thus extract discriminative activations through convolutional networks.

Moreover, due to the fact that internal representation is crucial for metric learning, in this paper, a feature aggregation module is applied for generating embeddings. As a widely used pooling method, global average pooling (GAP) could weaken the discrimination of the internal representation. And another popular pooling method, max pooling has its own disadvantage is that the valuable information of other parts in activations would be ignored. Instead of using GAP and max pooling methods, we applied a task-related module of which parameters can be designed for application scenarios to promote adaptation. Experimental results also show that the aggregation strategy has a substantial impact on the inter-class distance and intra-class distance in the metric space.

For the few-shot learning application and fair comparison of models, we only use descriptor selection module and feature aggregation module in fine-tuning and testing. The two modules allow meta-networks are easy and fast to fine-tune, and make significant progress on few-shot classification in two datasets. In effect, by utilizing images' own unsupervised information and the task-related aggregation, the internal representation discards the noise and background information and makes the adaption happen in the right space. Moreover, the descriptor selection module and feature aggregation module are compatible to most of existing meta-networks and can be applied in any stage (*i.e.* pre-training, meta-learning and testing), please refer to Section 4.5 for detail.

In conclusion, three contributions of UDS are as follows:

1. An unsupervised descriptor selection module is introduced to extract sample-related information in image, which removes irrelevant parts and promotes metric learning in adaptation.
2. We propose a task-related feature aggregation module to enhance internal representations in meta-learning, which generates compact embeddings and further improves network adaptation ability.
3. We conduct extensive experiments on datasets Caltech-UCSD Bird (CUB) and *mini*ImageNet. The experimental results demonstrate that with a simple structure, the proposed model obtains comparable performance and further promotes the classification accuracy when applied in prior meta-networks.

2. Related work

2.1. Few-shot learning

The studies of few-shot learning have been of interest for some time [15,16]. Previous probabilistic generative methods were only effective in the simple few-shot learning task, *e.g.* handwriting recognition. After embracing successful convolutional neural networks, Vinyals et al. [6] first introduced a training mechanism called episode, which mimics the few-shot learning tasks by utilizing sampled mini-batches during training. The use of episode makes the training stage more consistent with the testing and thereby improves the agent performance. After that, diverse approaches use this learning-to-learn or meta-learning mechanism for extracting transferrable knowledge from tasks. Current meta-learning methods can be divided following types:

Metric-learning based methods aim to learn an appropriate embedding or metric space and tackle the few-shot classification problem by comparing sample-to-sample or sample-to-class embedding distances. Siamese neural networks were first utilized in [5], which shares parameters in feature extraction and predicts whether two images belong to the same class. Matching network [6] calculates cosine distances between images and adopts a bidirectional LSTM in Full Context Embeddings (FCE) module to capture information. Snell et al. [10] uses $l2$-distance and compares similarity between images and prototypes of each class. In Relation network [17], the linear classifier is replaced by the Relation module which can directly learn the nonlinear-distance metric space. Nguyen et al. [18] equips Prototypical Network [10] with a novel dissimilarity measure that forces embeddings attracted to the correct prototypes and repelled from incorrect prototypes. As another improvement for Prototypical Network, Zhu et al. [19] proposes Temperature Network which generates query-specific prototypes and encourages compact intra-class distribution in metric space.

Memory based methods leverage memory-based neural networks, such as RNN and LSTM. Memory based methods aim to train a meta-learner to match new concepts and discard irrelevant information in memory. For example, Memory-augmented neural network [11] extracts knowledge from few samples through reading and writing external memory. Munkhdalai and Yu [12] introduces a memory-based parameterization approach to parameterize embedding function in meta-network. In order to leverage abundant unlabeled data, Zhu and Yang [8] introduces the label independent memory to facilitate few-shot classification in a semi-supervised way. Besides, Zhang et al. [7] constructs the cross-modal memory network where each slot stores visual and textual embedding pairs to provide multi-modal information.

Optimization based methods are trained to learn an optimizer or find an initialization of networks, induce great evaluation performance in the novel task within a few gradient-descent update steps. For instance, Ravi and Larochelle [13] replaces gradient descent optimizer with a LSTM-based meta-learner to learn the optimization in classifier training. Model Agnostic Meta-Learning (MAML) [14] is another representative model that aims to learn an initialization of the network, and the parameters can be adapted to new tasks through a few descent steps. To alleviate meta-overfitting in the optimization based methods, a network pruning approach is proposed in [20] to controlling the capacity of the over-parameterized meta-networks. Xu et al. [21] analyses the effect of data augmentation in the inner/outer training loop, and proposes an unsupervised meta-learning method which is learned from unlabeled datasets and further adapted to specific few-shot tasks.

2.2. Localization-based networks

While significant progresses have been made in few-shot classification, recent meta-learning methods have not achieve superior performance to human. When facing a new task, localizing and separating novel visual contents fit human intuition. The idea in [22] is inspirational and achieves great success in fine-grained image retrieval, but not suitable for meta-learning scenario since directly vector concatenation and layer ensemble could cause overfitting in high-dimensional space. Very recently, approaches [23–26] utilizing object localization or focusing regions have been proposed: To localize objects, [23] uses limited bounding annotations before classification; [24] adopts a saliency network to segment foregrounds and backgrounds, and combines them in a hallucination way to obtain augmented new samples; [25] employs class attention maps, which are trained by base categories to generate focus regions for enhancing feature presentation; [26] trains a triple-input module and a classifier based on image-to-class distance and obtains discriminative representations by weakly-supervised object localization. Current localization-based meta-networks are high-cost and hardly reusable, which need manual annotations of samples, or readily available localization, or mining data to generate

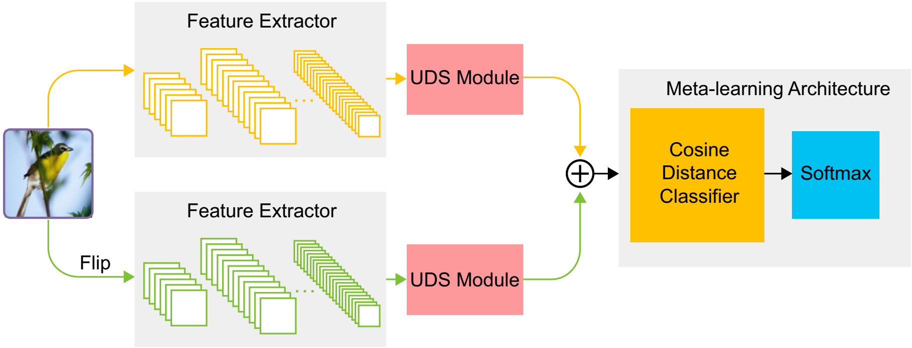


Fig. 1. The overall scheme of UDS in the standard setting. It mainly consists of three parts: (1) Pre-trained CNN networks to extract features of the original image and the flipped image. (2) Descriptor selection module to select semantic meaningful descriptors and aggregation module to generate embeddings. (3) Meta-learning architecture to perform few-shot classification with the input of the sum of two embeddings.

triplet-inputs. In contrast, the proposed UDS utilizes localization without supervision and achieves state-of-the-art results. Furthermore, UDS can be applied to most of existing models to improve few-shot classification performance. Considering few-shot learning application and dimension sensitivity of meta-networks, we propose a new descriptor selection module and a novel aggregation module in the UDS. UDS is an unsupervised method, which selects convolutional descriptors to obtain more discriminative representations of features. Thus, UDS makes meta-networks are easy and fast to fine-tune, allowing the adaption happen in the right space.

3. Methodology

3.1. Preliminary

In the setting of few-shot classification, we aim to learn an unsupervised descriptor selection based meta-network with few labelled samples. Like prior meta-learning approaches, we first pretrain a CNN backbone in training, and then learn a new classifier through meta-tasks in fine-tuning. In the training stage, feature extractor $f_\theta()$ and classifier $C_b()$ are trained with samples on base dataset $D_{base} = \{(x_i, y_i)\}_{i=1}^{M_{base}}$, where label y_i belongs to the base label set L_{base}. Parameters in $f_\theta()$ and $C_b()$ are updated by gradient descent in this stage.

In the fine-tuning stage, the parameter θ in the feature extractor $f_\theta()$ are fixed, and classifier $C_b()$ is replaced by a novel classifier $C_n()$. $C_n()$ is trained in episodes with given few samples on the novel dataset $D_{novel} = \{(x_j, y_j)\}_{j=1}^{M_{novel}}$, where label y_j belongs to the novel label set L_{novel}. Note that the image categories in L_{base} and L_{novel} are disjoint. For a N-way K-shot classification task, in each episode, we first randomly select N novel classes from L_{novel}. The support set $S_{novel} = \{(x_j, y_j)\}_{j=1}^{T_s}$ are sampled from D_{novel} and includes K samples of N novel classes, *i.e.*, $T_s = N \times K$. Query set Q_{novel} contains T_q samples belong to these N novel classes, and the network is optimized to minimizes the prediction loss of samples in L_{novel}.

The framework of UDS network for few-shot classification is illustrated in Fig. 1

3.2. Feature extractors

Various convolutional neural network settings have been used in previous methods to extract semantic information, but methods have complex performances in different depth of backbones. As mentioned in [27], increasing backbone depth is helpful to reduce intra-class variation and the gap among existing methods will be correspondingly reduced, but may cause performance decrease in some case. In this paper, different backbones are adopted with three depth: Conv-4, ResNet-10 and Resnet-34, which aims to verify the efficiency and stability of the proposed UDS.

We remove the final flatten layer in the CNN extractor $f_\theta()$, and add the descriptor selection module between the backbone and classifier, which is introduced in Section 3.3. In the training stage, we simply replace the flatten layer with a GAP layer instead. $f_\theta(x_i)$ maps an input image x_i to an 3-order Tensor $F = \{F_1(i,j), F_2(i,j), \ldots, F_k(i,j), \ldots, F_d(i,j)\}$ which includes $h \times w \times d$ elements. d is the number of feature maps and $h \times w$ is the spatial resolution of feature maps $F_k (k = 1, \ldots, d)$. F_k indicates the kth feature map of the corresponding channel, and $F(i,j)$ is a d-dimensional descriptor in position (i,j) of features.

3.3. Unsupervised descriptor selection

In the following, the unsupervised descriptor selection module will be introduced, and the feature aggregation module will be proposed in Section 3.4. Note that all processes in Section 3.3-3.4 are designed for real application in few-shot learning, which aims to transfer knowledge when only given a pre-trained CNN network and few cross-domain samples, *i.e.*, all these processes can be directly applied in fine-tuning.

Descriptor selection: Considering a pre-trained CNN feature extractor of which the last flatten layer is removed, the feature F of size $h \times w \times d$ can be obtained through the feature extractor. Each of feature maps in feature F can be treated as the processing result by one filter in the CNN. Observing activate values of feature maps, as shows in Fig. 2, we can infer semantic meaningful regions (highlighted by warm color) of the original image. However, backgrounds or noisy parts can be also extracted through the convolutional networks, which further obstruct learning in fine-tuning.

In order to focus meaningful regions and preserve sample-related information in the image, we first calculate the sum of the feature maps F_k through the channel direction, so the 3-order tensor feature F transforms into a 2-D activation map of size $h \times w$, *i.e.*, $F_{2-D} = \sum_{k=1}^{d} F_k$. Then the mask is generated by processing activations of the sample and removes backgrounds and noisy parts in features. The processing is similar to [22] which has achieved significant success on image retrieval. First, we calculate the average value of activation map F_{2-D}, given in Eq. (1):

$$Avg = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} F_{2-D}(i,j) \tag{1}$$

And then, Avg is the threshold to select useful descriptors from feature F. The descriptor $F(i,j)$ needs to be preserved if its corresponding element $F_{2-D}(i,j)$ is larger than Avg and discarded in contrast. The mask of features can be obtained by Eq. (2), and the

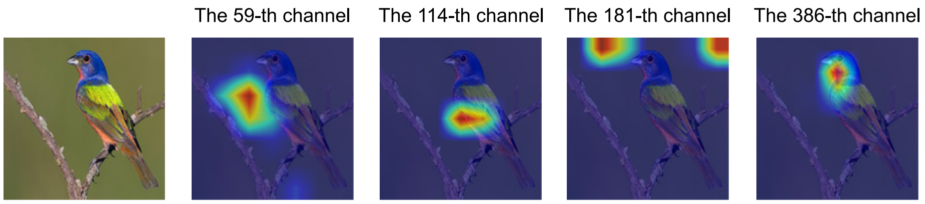


Fig. 2. Random sampled feature maps in different channels. To better visualization, we overlay feature maps to original image.

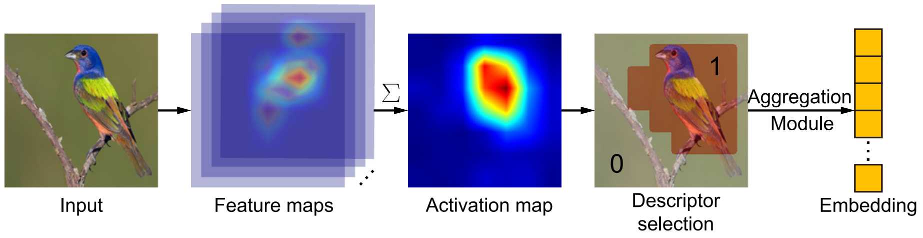


Fig. 3. A brief illustration of the descriptor selection module. It mainly selects descriptors through unsupervised localization and then output results for further aggregation.

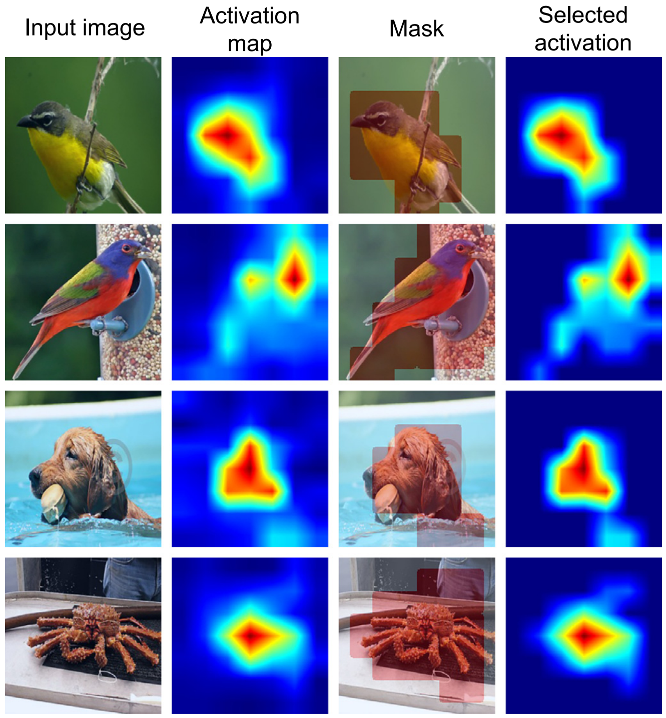


Fig. 4. Qualitative examples. From up to down, the first two rows are sampled from the CUB and the next two rows are sampled from *mini*ImageNet. For each of samples, the first column represents the original image; the second column represents the activations without descriptor selection; the third and fourth column represent the generated mask and the activations after selection.

descriptor $F(i,j)$ in the position (i,j) will be multiplied by 1 if the corresponding $Mask_{i,j} = 1$ or multiplied by 0 in contrast. The whole descriptor selection module is shown in Fig. 3.

$$Mask_{i,j} = \begin{cases} 1 & \text{if } F_{2-D}(i,j) > Avg \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

Qualitative examples: Fig. 4 shows activations comparison of examples in CUB and *mini*ImageNet. The mask generated from feature F has the size of $8 \times 8 \times 64$ after passing 4 convolution blocks. For better visualization, we resize the masks and activation maps by bilinear interpolation. As shown in Fig. 4, in activation maps existing irrelevant parts around the main objects, and the generated masks cover objects in original images. The masks indicate semantically meaningful regions in original images as birds, the dog and the crab are localized without supervision. Through the selection of descriptors, sample-related information is preserved and noises are removed, which makes activation maps clearer and more cohesive.

3.4. Aggregation strategy

Pooling: Metric-space learning is sensitive to the quality and dimensionality of embeddings. Therefore, choosing an efficient aggregation strategy is necessary after descriptor selection. GAP and max pooling are conventional pooling methods, however, the former will weaken the discrimination of the representations and the latter will lose the information of the indistinctive parts of feature maps. In the following, we will introduce Avg&max pooling and GeM pooling as they are the combinations of the two conventional pooling methods.

- **Avg&max pooling:** The pooling method separately calculates average and maximum values of each feature map, and concatenates results through the depth to get a $2 \times d$-dimensional embedding $l_{\mathrm{AVG\&MAX}}$, where

$$\begin{gathered} l_{\mathrm{AVG\&MAX}} = [f_1^{(a)} \dots f_k^{(a)} \dots f_d^{(a)}, f_1^{(m)} \dots f_k^{(m)} \dots f_d^{(m)}]^{\mathrm{T}}, \\ f_k^{(a)} = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} F_k(i,j), \; f_k^{(m)} = \max_{(i,j)} F_k(i,j) \end{gathered} \tag{3}$$

Due to the dimension sensitivity of meta-networks, expanding embedding dimension could cause overfitting in the meta-learning and further constrain the performance of classification.

- **GeM pooling:** Generalized-mean pooling (GeM Pooling) is widely exploited in image retrieval [28–30] and defined as

$$l_{GeM} = [f_1^{(g)} \dots f_k^{(g)} \dots f_d^{(g)}]^{\mathrm{T}},$$

$$f_k^{(g)} = \Big(\frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} F_k(i,j)^p\Big)^{\frac{1}{p}} \tag{4}$$

Similar with Avg&max pooling, GeM pooling incorporates GAP and max pooling methods, but keeps embeddings in d-dimensional space.

Embedding quality and dimension are critical to the metric-learning specially when data is severely limited in the few-shot learning task. So, GeM is adopted in the aggregation module and obtains excellent performance in few-shot learning. As shown in the aggregation strategy comparison in Section 4.5.2, GeM exceeds

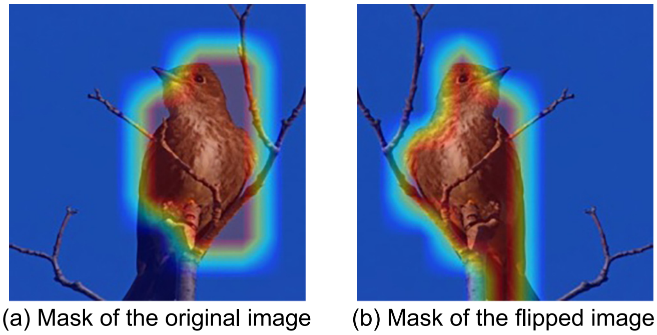

(a) Mask of the original image (b) Mask of the flipped image

Fig. 5. Mask comparison of the image and its flip version. To better visualization, we overlay masks to the corresponding image.

conventional pooling methods GAP and max pooling because they are special cases of GeM pooling, *i.e.*, when $p = 1$, GeM turns into GAP, and turns into max pooling when $p \to \infty$. Compared with Avg&max pooling, GeM improves the embedding of networks without increasing space dimensionality. Besides, the parameter p in GeM can be further adapted according to the real task and network setting, which is suitable in few-shot application. Unless otherwise stated, we set $p = 2$ for the convenience of comparison, and the ablation study of value p is shown in Section 4.5.3.

Image flip: Image flip is a self-ensemble strategy [31] in testing, which is widely used to enhance results in vision tasks [32,33]. In this paper, image flip is adopted in fine-tuning to enhance internal representations and avoid overfitting during metric-space learning. In fine-tuning, inputting an original image x_i and its horizontal flipped copy $\widetilde{x_i}$, distinct features and masks are obtained from backbone and descriptor selection module since CNN is not flip-invariant, as illustrated by Fig. 5. Then the processed features are mapped to two d-dimensional embeddings l_i and $\widetilde{l_i}$ by GeM pooling. l_i and $\widetilde{l_i}$ are incorporated into a new d-dimensional embedding e_i by element-wise addition, *i.e.*, $e_i = l_i + \widetilde{l_i}$. After that, the classifier C_n will be trained with embeddings e_i and corresponding labels in each episode. As mentioned above, image flip strategy makes image embeddings more robust and fuses the information of flipped images in meta-learning.

3.5. Cosine-distance classifier

Given d-dimensional embeddings of input images, the cosine-distance based classifier $C_n \in \mathbb{R}^{d \times N}$is used to predict categories of samples, written as

$$C_n = [w_1, w_2, \ldots, w_k \ldots, w_N], w_k \in [-1, 1]^d \tag{5}$$

where w_k indicates the weight vector of the kth class. For an input image x_i, we can get its embedding vector e_i and similarity scores $[s_{i,1}, s_{i,2}, \ldots, s_{i,N}]$ for all classes, where $s_{i,j} = e_i^\top w_j / \|e_i\| \|w_j\|$ means the similarity between embedding e_i and the jth class weight vector. Next, the prediction probability $P_{i,j}$ in the episode can be computed by the softmax function as

$$P_{i,j} = \frac{\exp(s_{i,j})}{\sum_{k=1}^{N} \exp(s_{i,k})} \tag{6}$$

Additionally, cross-entropy loss function is used to optimize the cosine-distance based classifier C_n in fine-tuning stage.

4. Experiments

In this section, extensive experiments are conducted on three datasets to validate the effectiveness and superiority of the UDS method. The overall scheme of our method as shown in Fig. 1.

4.1. Datasets

The proposed UDS is evaluated on two datasets Caltech-UCSD Bird (CUB) and *mini*ImageNet. Their details are as follows:

- **CUB** [34]: The dataset is a well-known fine-grained image dataset for few-shot classification. It consists of 11,788 images belong to 200 different bird species. We split the dataset into 100 base, 50 validation and 50 novel classes, closely follow the procedure of Hilliard et al. [35].
- ***mini*ImageNet** [6]: The dataset is a widely used generic object recognition dataset. As a mini-version of the ImageNet [36], *mini*ImageNet is first proposed by Vinyals et al. [6] to address few-shot learning problem. *mini*ImageNet consists of 100 classes where each class contains 600 examples and we randomly select 64 base, 16 validation and 20 novel classes like the work in [13].

4.2. Competitors

We compare our UDS with ten existing state-of-the-art methods to fully demonstrate its effectiveness and superiority. Details of competitors are introduced as follows.

- **Matching Network** [6]: This method introduces meta-learning mechanism that maps images in support set and query set into embedding space. Bidirectional LSTM is also applied to adjust representations after feature extraction, then the cosine distance between query and examples of each class will be calculated to perform few-shot learning.
- **Prototypical Network** [10]: Different from Matching Network, Prototypical Network evaluates $l2$-distance similarity between query and class prototypes in embedding space. The prototypes are represented by the mean of embeddings in each category.
- **Relation Network** [17]: This method designs a CNN-based relation module which measures similarities between query and support samples. Relation Network is an end-to-end model that learns feature embedding in training and directly learns a non-linear distance metric.
- **Model-Agnostic Meta-learning (MAML)** [14]: This method is an optimization based meta-learning method, which allows the network to deal with the scenario of few training samples. MAML aims to learn a great initialization of the neural network that can adapt new learning tasks through a few descent steps.
- **Baseline** [27]: Baseline is a meta-learning method with the standard setting, which contains a pre-trained backbone and a cosine-distance based classifier. Baseline achieves competitive performance in meta-tasks, and provides a unified testbed to fairly compare our model and methods mentioned above.
- **Feature Fusion** [25]: This method utilizes focus-areas of the image to enhance feature representations. Localization in this method is generated by Grad-CAM [37]. Grad-CAM is learned from samples in base category, and fixed when handling novel meta-tasks.
- **Few-shot Learning with Weakly-supervised Object Localization** [26]: Similar to Feature Fusion method, Grad-CAM is also adopted in this method. The method first mines triplet-inputs in samples, and applies an Image-To-Class-Distance (ITCD) localizer to obtain discriminative regions with weakly supervision.
- **Few-Shot Learning with Localization in Realistic Settings** [23]: This method utilizes bounding box annotations to predict the foreground and background of images without annotation. Then, the predicted foreground and background vectors will be concatenated to perform few-shot classification.
- **Deep Nearest Neighbor Neural Network (DN4)** [38]: This method replaces conventional image-level embedding distance

with image-to-class local descriptor distance. DN4 calculates similarities between each local descriptor in query and all descriptors which belong to one class in support set, then adds k-nearest neighbors of similarities together as the image-to-class similarity.

- **Target-Oriented Alignment Network (TOAN)** [39]: This method proposes Target-oriented Matching Mechanism (TOMM) and Group Pair-wise Bilinear Pooling (GPBP). The former module reformulates spatial features to reduce intra-class variance in the embedding space, and the latter aims to enhance inter-class discrimination by incorporating local compositional information.

Note the Matching Net, Prototypical Net, Relation Net, MAML and Baseline are conventional meta-learning methods, which are re-implemented in the same testbed and have minor setting differences between them. Additionally, we use a first-order approximation version of MAML for memory efficiency. The other five competitors hold a similar view like us and utilize objects localization in meta-learning. Compared with these methods, object localization in our UDS is unsupervised and without annotation.

4.3. Implementation details

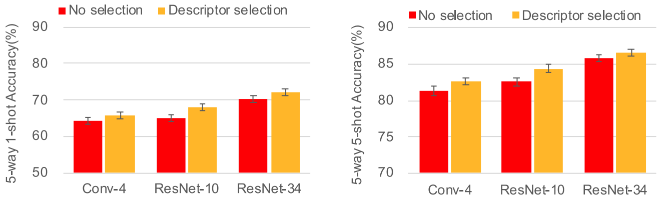
Since implementation details in meta-learning methods are different, we implement our model in a unified meta-learning testbed [27] to assess the effectiveness of UDS. The testbed contains a baseline and several conventional meta-learning methods mentioned above. Since the various backbones used by existing localization-based methods bring difficulties for comparison, three backbones Conv-4, ResNet-10 and ResNet-34 are adopted to UDS in order to compare performance as fairly as possible. The details of Conv-4 can refer to [10] with an input size of 84 × 84, ResNet-10 and ResNet-34 with an input size of 224 × 224 as followed in [27] and [40] respectively. Besides, data augmentation is applied in both of training and fine-tuning, which includes random crop, horizontal flip and color jitter.

We first train the feature extractor $f_\theta()$ in the training stage, then fix the parameter θ of extractor $f_\theta()$ to learn a cosine-metric classifier $C_b()$ in fine-tuning. Unless otherwise stated, the descriptor selection module and aggregation module are only applied in fine-tuning to evaluate the performance for few-shot learning. As described in Section 3.3, we do not apply two modules in training stage and correspondingly replace them with a GAP layer. In the training stage, a feature extractor is learned by using Adam optimizer with the learning rate 10^{-3}, and we train 200 epochs in CUB and 400 epochs in *mini*ImageNet with a batch size of 16. In the fine-tuning stage, we randomly sample 60,000 episodes to train the cosine-distance based classifier both in 1-shot learning and 5-shot learning. Additionally, we fix parameter $p = 2$ in aggregation module for convenience and the cosine-distance classifier is multiplied by 4 as the fixed scalar to promote the gradient back-propagation. In the testing stage, the prediction accuracy is averaged over 600 experiments. In each of experiment, we sample 5 classes from the novel class set L_{novel}, and dividedly pick k samples for the support set and 16 for the query set in each class.

4.4. Few-shot classification performance

Under above settings, we perform experiments on CUB and *mini*ImageNet. By comparing the results of CUB and state-of-the-art methods in Table 1, we have the following observations:

- We notice that the backbone depth employed by meta-network has a direct impact on the classification accuracy. For example, on CUB, the performance of UDS in 1-shot classification obtains 65.84%, 68.05% and 72.23% with backbones Conv-4, ResNet-10 and ResNet-34 respectively. In most cases, deeper backbone means higher performance of models, but increasing backbone depth is not always helpful to meta-learning, as further results shown in Section 4.5.4.



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

Fig. 6. Ablation on descriptor selection.

- In most cases, the localization-based methods perform better than conventional meta-learning methods and baseline. That indicates utilizing localization effectively improves meta-learning performance by extracting semantic meaningful parts in the image.
- The proposed UDS model consistently achieves the best accuracy on CUB, even when its backbone is shallower than other competitors. This result proves localizing without supervision is also effective in fine-grained few-shot classification. Moreover, enhancing internal representations by aggregation module is helpful to learn a compact embedding space and thus obtain better generalization ability.
- For *mini*ImageNet, UDS obtains comparable performance to conventional meta-learning methods, but loses to FSWL in shallow backbone Conv-4. It is reasonable that FSWL utilizes weakly-supervised localization through triplet-inputs, and localization in this paper is unsupervised. Besides, the limited capacity of backbone and complex scenarios in *mini*ImageNet obscure unsupervised descriptor selection. When the backbone getting deeper, the accuracy of UDS improves 9.47% in 1-shot classification and 9.48% in 5-shot classification. We can see that the performance gap drastically reduces and UDS remains the most competitive one.

4.5. Ablation study

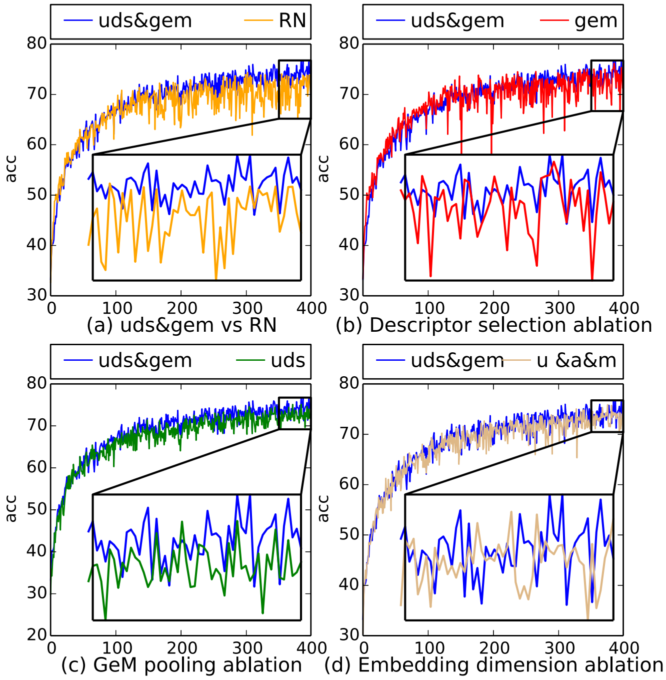
In this section, we perform ablation studies to compare the individual contribution of each component in UDS. We first briefly study the effectiveness of descriptor selection in Section 4.5.1, and separately investigate the difference of variants of aggregation strategy in Section 4.5.2. Then, the influence of parameter p in GeM pooling are evaluated in Section 4.5.3. In Section 4.5.4, two modules in UDS will be applied to four well-known meta-learning methods to further prove their effectiveness and compatibility. Unless otherwise stated, all experiments are conducted by the same setting with the fixed seed on CUB, and equipped with three backbones: Conv-4, ResNet-10 and ResNet-34.

4.5.1. Descriptor selection analysis

In fine-tuning, for any feature F extracted by CNN, we utilize the descriptor selection module to select semantic meaningful regions and remove noisy parts in images. In this section, we study the influence of the descriptor selection module on 5-way 1-shot and 5-shot classification tasks. We separately test the performance of our model with and without descriptor selection and report the results in Fig. 6. For any backbone depth, the performance with descriptor selection performs better than normal fine-tuning scheme without selection. The result demonstrates that focusing semantic meaningful regions and removing backgrounds are beneficial to metric-space learning in fine-tuning stage.

Table 1
5-way accuracy (%) of all methods on CUB and *mini*ImageNet, with 95% confidence intervals. The second column refers to the backbone employed by the method. FSWL indicates Few-shot Learning with Weakly-supervised Object Localization, and FSL means Few-Shot Learning with Localization in Realistic Settings. † denotes conventional meta-learning methods re-implemented in similar settings as UDS. Other results are retrieved from the original work.

Method	Backbone	CUB		miniImageNet	
		1-shot	5-shot	1-shot	5-shot
Matching Net† [6]	Conv-4	61.93 ± 0.91	75.56 ± 0.72	47.37 ± 0.80	61.89 ± 0.71
Prototypical Net† [10]	Conv-4	51.45 ± 0.91	74.51 ± 0.69	43.17 ± 0.79	63.11 ± 0.72
Relation Net† [17]	Conv-4	63.54 ± 0.99	78.67 ± 0.67	46.53 ± 0.83	63.99 ± 0.71
MAML† [14]	Conv-4	60.96 ± 0.96	76.61 ± 0.69	47.42 ± 0.82	63.04 ± 0.71
Baseline† [27]	Conv-4	58.12 ± 0.85	77.23 ± 0.64	46.74 ± 0.74	66.05 ± 0.67
FSWL [26]	Conv-4	43.30 ± 0.75	62.21 ± 0.73	**53.30** ± **0.32**	**73.22** ± **0.45**
TOAN [39]	Conv-4	65.34 ± 0.75	80.43 ± 0.60	-	-
DN4 [38]	Conv-4	53.15 ± 0.84	81.90 ± 0.60	51.24 ± 0.74	71.02 ± 0.64
UDS (ours)	Conv-4	**65.84** ± **0.91**	**82.62** ± **0.55**	46.53 ± 0.75	66.19 ± 0.68
TOAN [39]	ResNet-12	67.17 ± 0.81	82.09 ± 0.56	-	-
Feature Fusion [25]	ResNet-18	47.89 ± 0.00	61.30 ± 0.00	-	-
UDS (ours)	ResNet-10	**68.05** ± **0.92**	**84.35** ± **0.53**	**56.00** ± **0.80**	**75.67** ± **0.61**
FSL [23]	ResNet-50	-	-	49.64 ± 0.31	69.45 ± 0.28
UDS (ours)	ResNet-34	**72.23** ± **0.88**	**86.57** ± **0.50**	**55.04** ± **0.81**	**75.11** ± **0.63**



(a) uds&gem vs RN (b) Descriptor selection ablation (c) GeM pooling ablation (d) Embedding dimension ablation

Fig. 7. Ablation comparison studies of UDS. Accuracy rates are obtained in 5-way 5-shot few-shot learning tasks on Relation with backbone Conv-4. In each sub-figure, the horizontal axis is the training epoch and the vertical axis represents the accuracy rate. "RN" represents the modified Relation Net, "uds" represents Relation Net with unsupervised descriptor selection, "gem" represents Relation Net with GeM pooling, "uds&gem" represents Relation Net employed with two UDS modules and "u&a&m" represents Relation Net applied with unsupervised descriptor selection and Avg&max aggregation mentioned in Section 3.4.

4.5.2. Aggregation strategy analysis

In the proposed model, the original image and its flip version are separately mapped to two embeddings through GeM pooling. Then the two embeddings will be added together and projected into a novel cosine-distance space. Apparently, the aggregation strategy and embedding dimension d_l have direct impacts on the performance of meta-networks. In this section, we compare our aggregation module with three pooling methods mentioned in Section 3.4 and aggregation strategies which utilize Avg&max pooling, image flip and multiple layer ensemble. The last three aggregation strategies generate $2 \times d_l$-dimensional, $4 \times d_l$-dimensional and $8 \times d_l$-dimensional embeddings respectively. For convenience, we fix parameter $p = 2$ in GeM pooling, and the influence of p on UDS will be investigated in Section 4.5.3. All 1-shot and 5-shot tasks are based on Conv-4 which contains 64 filters in the last convolution layer. As illustrated by Table 2, the proposed aggregation module outperforms GAP and max pooling since it can incorporate more information of feature maps. Compared with Avg&max pooling and its variants, the proposed aggregation module has better performance and does not increase dimensionality of embeddings, as the performance decline dramatically when space dimensionality increasing to $8 \times d_l$. The results indicate that the appropriate aggregation strategy and dimensionality of metric space are helpful to enhance internal representations and avoid overfitting, further improve adaptation ability in meta-learning.

Table 2
Comparison of different aggregation strategies on UDS. $^+$ denotes embedding concatenate the flipped image's embedding, $^{++}$ indicates embedding concatenate the flipped image's embedding and use multiple-layer ensemble as in [22]. Average 5-way accuracy (%) over 600 test episodes is reported with 95% confidence intervals.

Pooling	dimension	1-shot	5-shot
Average	64	65.12 ± 0.91	81.52 ± 0.57
Max	64	65.06 ± 0.90	82.34 ± 0.56
GeM	64	**66.23** ± **0.91**	**83.41** ± **0.53**
Avg&max	128	65.91 ± 0.92	82.51 ± 0.56
Avg&max$^+$	256	41.28 ± 0.83	60.96 ± 0.81
Avg&max^{++}	512	31.39 ± 1.05	22.65 ± 0.85

4.5.3. Impact of hyper-parameter p

To investigate the performance of different values of parameter p in Eq. (4), we perform 1-shot and 5-shot learning tasks by varying the value p from {1, 2, 3, 4, 5}. As shown in Table 3, parameter p has a mild impact on accuracy that proves our approach is robust to various p, with even higher accuracy at $p = 4$ than results reported above. Therefore, p in UDS should be selected based on the specific task in the real applications.

4.5.4. Compatibility analysis

Since our UDS is unsupervised and independent of meta-network architecture, we apply the proposed descriptor selection module and aggregation module on conventional meta-learning methods: Matching Net [6], Prototypical Net [10], Relation Net [17] and MAML [14]. Matching Net, Prototypical Net and Relation

Table 3
Ablation comparison studies of p value. Average 5-way accuracy (%) over 600 test episodes is reported with 95% confidence intervals.

p	Conv-4		ResNet-10		ResNet-34	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
1	65.19 ± 0.92	81.62 ± 0.56	67.08 ± 0.92	83.08 ± 0.54	70.80 ± 0.88	85.30 ± 0.52
2	**65.84 ± 0.91**	82.62 ± 0.55	68.05 ± 0.92	84.35 ± 0.53	72.23 ± 0.88	86.57 ± 0.50
3	65.60 ± 0.91	**82.67 ± 0.55**	68.13 ± 0.92	84.50 ± 0.53	72.36 ± 0.88	86.83 ± 0.49
4	65.37 ± 0.91	82.59 ± 0.56	**68.14 ± 0.91**	**84.57 ± 0.53**	**72.38 ± 0.88**	86.87 ± 0.49
5	65.25 ± 0.91	82.56 ± 0.56	68.07 ± 0.91	**84.57 ± 0.53**	72.37 ± 0.88	**86.91 ± 0.49**

Table 4
The five-way few-shot classification performance with and without UDS. * denotes UDS is applied in testing with/without fine-tuning, and ** indicates UDS is applied both in training and testing with/without fine-tuning. Average 5-way accuracy (%) over 600 test episodes is reported with 95% confidence intervals.

Methods	Conv-4		ResNet-10		ResNet-34	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet	61.93 ± 0.91	75.56 ± 0.72	71.68 ± 0.93	85.04 ± 0.57	73.18 ± 0.91	85.34 ± 0.57
MatchingNet*(ours)	65.96 ± 0.95	77.58 ± 0.69	74.22 ± 0.89	86.00 ± 0.53	72.70 ± 0.92	85.41 ± 0.54
MatchingNet**(ours)	69.52 ± 0.95	83.18 ± 0.60	75.19 ± 0.87	86.84 ± 0.51	72.89 ± 0.93	87.30 ± 0.52
ProtoNet	51.45 ± 0.91	74.51 ± 0.69	72.86 ± 0.95	84.75 ± 0.56	72.39 ± 0.95	87.18 ± 0.48
ProtoNet*(ours)	58.38 ± 0.97	75.56 ± 0.69	71.32 ± 0.92	84.91 ± 0.51	71.36 ± 0.91	78.88 ± 0.61
ProtoNet**(ours)	62.37 ± 0.96	79.99 ± 0.63	74.41 ± 0.92	86.70 ± 0.51	72.29 ± 0.95	87.65 ± 0.48
RelationNet	63.54 ± 0.99	78.67 ± 0.67	69.60 ± 1.00	83.09 ± 0.61	67.96 ± 1.00	83.48 ± 0.58
RelationNet(ours)	62.67 ± 1.01	76.43 ± 0.72	68.90 ± 0.99	82.04 ± 0.62	69.99 ± 0.99	84.13 ± 0.59
RelationNet*(ours)	60.28 ± 0.99	75.68 ± 0.71	71.02 ± 0.99	83.58 ± 0.60	70.80 ± 0.97	84.53 ± 0.56
RelationNet**(ours)	63.54 ± 0.99	78.48 ± 0.72	72.21 ± 0.96	84.11 ± 0.58	70.68 ± 0.96	85.28 ± 0.56
MAML	56.58 ± 0.98	74.86 ± 0.73	73.02 ± 0.96	80.08 ± 0.74	69.28 ± 1.02	82.57 ± 0.61
MAML*(ours)	59.92 ± 1.04	67.41 ± 0.95	66.44 ± 0.98	77.80 ± 0.85	67.31 ± 1.06	80.95 ± 0.66
MAML**(ours)	64.68 ± 0.99	68.94 ± 0.80	72.06 ± 1.00	81.62 ± 0.68	70.41 ± 1.01	82.51 ± 0.63

Net are metric-learning based methods and trained with episodes sampled from base dataset D_{base}. Since there is no fine-tuning process in their original work, we correspondingly add UDS in the testing. MAML aims to learn an initialization of networks and parameters will be adapted to new tasks in fine-tuning. Therefore, UDS is further applied to the fine-tuning in MAML. The basic setting of models is the same as described in Section 4.4.

In order to apply UDS to Matching Net, Prototypical Net and MAML, in testing with/without fine-tuning, the final layer of backbone is replaced by the descriptor selection module, then the aggregation module is applied in those methods. As for Relation Net, UDS cannot be directly applied since the backbone is connected with the CNN-based relation modules. In original Relation Net, the 3-order tensor processed by relation modules will be flattened, and then sent to a MLP to predict relation scores. Accordingly, we plan to add UDS to the last relation module instead of backbone in the test stage of Relation Net. For this reason, flatten layer is replaced by a GAP in the modified Relation Net. In addition, a FC layer is correspondingly added to MLP to maintain the network capacity, since the input dimension of MLP is reduced by GAP. As shown in Table 4, the modified Relation Net achieved similar performance to the original work. In the testing stage of the modified Relation Net, in order to apply UDS, descriptor selection module is connected to the last relation module and the GAP layer is replaced by GeM pooling.

5-way classification results (denoted by *) on three backbones are reported in Table 4. While Matching Net, Prototypical Net, Relation Net and MAML have complicated performance on all of tasks, UDS boosts the performance of meta-learning methods in most cases. UDS directly extracts semantic information from features and boosts the performance a great margin specially when samples and network capacity are limited, such as 1-shot classification in Conv-4. Few-shot learning results prove that UDS is effective to improve metric-learning methods' performance when applied in testing with/without fine-tuning.

Moreover, we further adopt UDS in the training stage (denoted by **), as shown in Table 4, the few-shot classification performance boosts by a large margin. The potential reason is that UDS assists feature extraction during training, which helps network to learn a more discriminative extractor. Results indicate that UDS can be further applied in training stage to learn a more effective feature extractor when meta-network is trained from scratch. Take Relation Net for example, we compare the individual contributions of different components in UDS. We show the performance curves of Relation Net variants for 5-way 5-shot learning in Fig. 7. We find that the Relation Net equipped with two UDS modules consistently exceeds the Relation Net curve in Fig. 7(a). The result demonstrates that UDS prompt the training of Relation Net. We separately test the performance of descriptor selection and GeM pooling in Relation Net. By comparing results shown in Fig. 7(b) and (c), we conclude following points. First, the ablation of descriptor selection or GeM pooling results in performance decrease. Therefore, both the modules are helpful to the few-shot classification. Second, curves are sharper without descriptor selection or GeM. It indicates that network equipped with two modules is more robust, which avoids performance dramatically decrease in some case. Third, it is interesting to note that the network equipped with both modules minor loses network only using GeM at epoch 1–150. This situation is reasonable because the feature extraction is not effective at the beginning, which obstructs the selection of descriptors since the mask is generated from the activation map. We also investigate the influence of embedding dimension by using Avg&max pooling and results are shown in Fig. 7(d). As described in Section 3.4, compact embedding performs better in few-shot learning (GeM pooling generates c-dimensional embeddings and Avg&max pooling generates $2 \times c$-dimensional embeddings, here c is the channel number of the last relation module).

5. Conclusions

In this paper, we propose the UDS model for better performing few-shot learning, which includes two modules: the descriptor selection module and the aggregation module. The feature maps are utilized to select semantic meaningful descriptors, and embeddings are generated by a more effective aggregation to learn a robust metric space. Unlike prior localization-based meta-learning methods, UDS model is unsupervised and does not complicate the network structure or need extra sample mining for localization. We conduct extensive experiments on the fine-grained image dataset CUB and generic object recognition dataset *mini*ImageNet. The results validate the superiority of the UDS and the compatibility of the proposed modules in prior meta-networks. At present, the localization obtained from a single image is not precise enough to separate the object, and we will investigate how to better employ the co-localization of support samples belong to the same category, thus further improving few-shot classification performance. Additionally, the ideas adopted by this work could also benefit other few-shot tasks. For example, besides the general improvements for few-shot classification, when performing activity recognition, further improvements are likely obtained by refining internal representations with UDS. Also, the unsupervised localization of this paper may also be beneficial to few-shot object detection or tracking tasks, as networks can be learned from unlabeled datasets and further adapted to the specific few-shot tasks. We plan to investigate these topics in the future work.

Declaration of Competing Interest

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

Acknowledgement

This work was supported by the National Natural Science Foundation of China [grant number 61771420] and the Natural Science Foundation of Hebei Province [grant number F2016203422]. We thank Yu Han for his linguistic assistance during the preparation of this manuscript. In this paper, we utilize the public few-shot learning baseline and thank the provider of the baseline.

References

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.

[2] M. Zhou, N. Duan, S. Liu, H.-Y. Shum, Progress in neural NLP: modeling, learning, and reasoning, Engineering 6 (3) (2020) 275–290.

[3] J. Wu, S. Zhang, X. Li, J. Chen, H. Xu, J. Zheng, Y. Gao, Y. Tian, Y. Liang, R. Ji, Joint segmentation and detection of COVID-19 via a sequential region generation network, Pattern Recognit. 118 (2021) 108006.

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

[5] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, in: Proceedings of the ICML Deep Learning Workshop, vol. 2, 2015.

[6] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: Advances in Neural Information Processing Systems (NIPS), 2016, pp. 3630–3638.

[7] L. Zhang, X. Chang, J. Liu, M. Luo, M. Prakash, A.G. Hauptmann, Few-shot activity recognition with cross-modal memory network, Pattern Recognit. 108 (2020) 107348.

[8] L. Zhu, Y. Yang, Label independent memory for semi-supervised few-shot video classification, IEEE Trans. Pattern Anal. Mach.Intell. (PAMI) (2020). 1–1

[9] Z. Ji, X. Chai, Y. Yu, Y. Pang, Z. Zhang, Improved prototypical networks for few-shot learning, Pattern Recognit. Lett. 140 (2020) 81–87.

[10] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems (NIPS), 2017, pp. 4080–4090.

[11] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: Proceedings of the International Conference on Machine Learning (ICML), 2016, pp. 1842–1850.

[12] T. Munkhdalai, H. Yu, Meta networks, in: Proceedings of the International Conference on Machine Learning (ICML), 2017, pp. 2554–2563.

[13] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[14] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the International Conference on Machine Learning (ICML), 2017, pp. 1126–1135.

[15] L. Fe-Fei, et al., A Bayesian approach to unsupervised one-shot learning of object categories, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2003, pp. 1134–1141.

[16] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach.Intell. (PAMI) 28 (4) (2006) 594–611.

[17] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H.S. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1199–1208.

[18] V.N. Nguyen, S. Løkse, K. Wickstrøm, M. Kampffmeyer, D. Roverso, R. Jenssen, SEN: a novel feature normalization dissimilarity measure for prototypical few-shot learning networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 118–134.

[19] W. Zhu, W. Li, H. Liao, J. Luo, Temperature network for few-shot learning with distribution-aware large-margin metric, Pattern Recognit. 112 (2021) 107797.

[20] H. Tian, B. Liu, X.-T. Yuan, Q. Liu, Meta-learning with network pruning, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 675–700.

[21] H. Xu, J. Wang, H. Li, D. Ouyang, J. Shao, Unsupervised meta-learning for few-shot learning, Pattern Recognit. 116 (2021) 107951.

[22] X.-S. Wei, J.-H. Luo, J. Wu, Z.-H. Zhou, Selective convolutional descriptor aggregation for fine-grained image retrieval, IEEE Trans. Image Process. 26 (6) (2017) 2868–2881.

[23] D. Wertheimer, B. Hariharan, Few-shot learning with localization in realistic settings, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6558–6567.

[24] H. Zhang, J. Zhang, P. Koniusz, Few-shot learning via saliency-guided hallucination of samples, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 2770–2779.

[25] X. Sun, H. Xv, J. Dong, H. Zhou, C. Chen, Q. Li, Few-shot learning for domain-specific fine-grained image classification, IEEE Trans. Ind. Electron. 68 (4) (2021) 3588–3598.

[26] X. He, J. Lin, J. Shen, Weakly-supervised object localization for few-shot learning and fine-grained few-shot learning (2020). arXiv preprint arXiv:2003.00874

[27] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C.F. Wang, J.-B. Huang, A closer look at few-shot classification, in: Proceedings of the International Conference on Learning Representations (ICLR), 2019.

[28] F. Radenović, G. Tolias, O. Chum, Fine-tuning CNN image retrieval with no human annotation, IEEE Trans. Pattern Anal. Mach.Intell. (PAMI) 41 (7) (2019) 1655–1668.

[29] J. Revaud, J. Almazán, R.S. Rezende, C.R.d. Souza, Learning with average precision: training image retrieval with a listwise loss, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 5106–5115.

[30] B. Xu, L. He, X. Liao, W. Liu, Z. Sun, T. Mei, Black re-ID: a head-shoulder descriptor for the challenging problem of person re-identification, in: Proceedings of the ACM International Conference on Multimedia, 2020, pp. 673–681.

[31] R. Timofte, R. Rothe, L. Van Gool, Seven ways to improve example-based single image super resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1865–1873.

[32] A. Abdelhamed, M. Afifi, R. Timofte, M.S. Brown, NTIRE 2020 challenge on real image denoising: dataset, methods and results, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2077–2088.

[33] S. Yuan, R. Timofte, A. Leonardis, G. Slabaugh, NTIRE 2020 challenge on image demoireing: methods and results, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1882–1893.

[34] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[35] N. Hilliard, L. Phillips, S. Howland, A. Yankov, C.D. Corley, N.O. Hodas, Few-shot learning with metric-agnostic conditional embeddings (2018). arXiv preprint arXiv:1802.04376

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.

[37] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.

[38] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, J. Luo, Revisiting local descriptor based image-to-class measure for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7253–7260.

[39] H. Huang, J. Zhang, L. Yu, J. Zhang, Q. Wu, C. Xu, TOAN: target-oriented align-

ment network for fine-grained image categorization with few labeled samples, IEEE Trans. Circuits Syst. Video Technol. (2021). 1–1
[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

Zhengping HU received the B.S. degree in wireless technology in 1996, the M.S. degree in circuit and system in 1999 from Yanshan University, and the Ph.D. degree in information and communication in 2007 from Harbin Institute of Technology, China. He is currently a professor in Yanshan University, and the dean of the Department of electronic and communication engineering. His research areas include deep learning, pattern recognition, video analysis, and meta-learning.

Zijun Li received the B.S. degree from the School of Mathematic and Statistics from Shandong Normal University, China, in 2019. Currently he is pursuing the M.S. degree in the School of Information Science and Engineering, Yanshan University, China. His research interests include machine learning, computer vision and few-shot learning.

Xueyu Wang is a master student at School of Information Science and Engineering, YanShan University, China. She is conducting research in video salient object detection, video instance segmentation and machine learning.

Saiyue Zheng is a master student in the School of Information Science and Engineering, YanShan University, China. Her research interests include metric learning and person re-identification.