SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2021, Vol. 64 192105:1–192105:13
https://doi.org/10.1007/s11432-020-2973-7

Learning to focus: cascaded feature matching network for few-shot image recognition

Mengting CHEN[1], Xinggang WANG[1], Heng LUO[2], Yifeng GENG[2] & Wenyu LIU[1*]

[1]*School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[2]*Horizon Robotics, Beijing* 100080*, China*

Received 27 January 2020/Revised 27 April 2020/Accepted 2 July 2020/Published online 30 July 2021

Abstract Generally, deep networks learn to recognize a category of objects by training on a large number of annotated images accurately. However, a meta-learning problem known as a low-shot image recognition task occurs when a few images with annotations are available for learning a recognition model for a single category. Consequently, the objects in testing/query and training/support image datasets are likely to vary in terms of size, location, style, and so on. In this paper, we propose a method, cascaded feature matching network (CFMN), to solve this problem. We train the meta-learner to learn a more fine-grained and adaptive deep distance metric using feature matching block, which aligns associated features together and naturally ignores non-discriminative features. By applying the proposed feature matching block in different layers of the network, multi-scale information among the compared images is incorporated into the final cascaded matching feature, which boosts the recognition performance and generalizes better by learning on relationships. Moreover, the experiments for few-shot learning (FSL) using two standard datasets: miniImageNet and Omniglot, confirm the effectiveness of our proposed method. Besides, the multi-label few-shot task is first studied on a new data split of the COCO dataset, which further shows the superiority of the proposed feature matching network when performing the FSL in complex images.

Keywords few-shot learning, image recognition, feature matching, self-attention

Citation Chen M T, Wang X G, Luo H, et al. Learning to focus: cascaded feature matching network for few-shot image recognition. Sci China Inf Sci, 2021, 64(9): 192105, https://doi.org/10.1007/s11432-020-2973-7

1 Introduction

Deep learning achieves great success in a variety of tasks with large amounts of labeled data for image recognition [1–3], machine translation [4, 5], and speech synthesis [6]. However, the labeled data is not always available massively when annotation cost is high or time is not allowed. Conversely, humans learn novel concepts with only a few examples in a short time [7].

The few-shot learning (FSL) attempts to solve this problem by training a model that classifies unlabeled examples using a small labeled support set. Specifically, N-way K-shot learning is the task of classifying an example, labeled as a query, into one of the N classes, when only K samples per class are available as supervision. These $N \times K$ samples with labels are labeled as a support set. During the training, the support images and some query images are sampled. The meta-learner distinguishes the category of query images using the support images only. Moreover, the categories of the training set different from the testing set are randomly sampled to avoid direct semantic relationships and visual similarities. The batch of the support set and queries is termed as an episode [8].

Given a test image, the FSL model estimates the feature similarities between the test and the supporting images of each class. Then, different from the traditional image recognition task (i.e., each class is represented by a parametric model learned from a large number of images), the category is supported by only a few, even a single image in the few-shot setting. It suggests that the classifier needs to accurately evaluate the similarity with little supervision and strong variance due to a lack of enough

* Corresponding author (email: liuwy@hust.edu.cn)

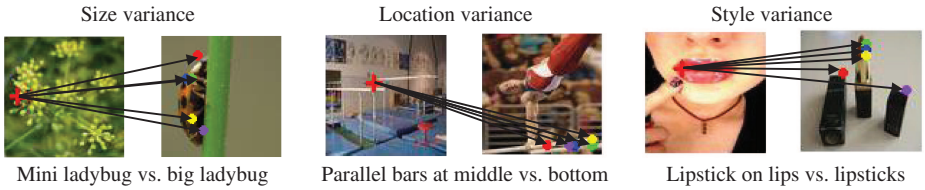

Figure 1 (Color online) Visualization of the feature matching results of the cascaded feature matching network (CFMN). Two adjacent images form a group. The feature at the red cross in the left query image matches with all features at the colored positions in the right support image. The colors: red, green, blue, yellow, and purple represent the top five highest correlation responses. Although the interesting objects may be different in size, location, style, they are associated together by our feature matching operation.

supporting information. As shown in Figure 1, the query image may share limited visual similarities with the support images. However, it is challenging for the model to generalize among the great inter-class differences with a limited number of training images per class. According to [9, 10], the FSL problem demonstrates overfitting. To solve the problem of great inter-class differences, we propose that the meta-learner should focus on essential spatial relationship features that correlate the query and support images and pay less attention to the non-discriminative features. Thus, we allow our feature matching block to align the features of the two compared images using the similarity of every feature position pair. As shown in Figure 1, the two positions corresponding to the object from the same category record a high response using our method, though the overall images may look quite different visually.

To utilize the proposed feature matching block fully, we apply three blocks at different layers of the network and cascade them together. The representation level of features from shallow layers of the convolution neural network (CNN) is different and usually lower than the deep layers CNN. The relation and similarity information of edges, shapes, and colors using shallow layers are extracted, whereas deeper layers can produce object parts or other semantic information. The cascaded structure fuses all the information to make the final decision more accurate and robust.

In this paper, however, our main contributions reflect the following five aspects: (1) We propose a feature matching block that is capable of associating the object parts with high correlations among compared images. The block enables the model to pay more attention to the parts, which generalize considerable intra-class variation between the query and support images for the FSL challenge. (2) The feature matching block is cascaded to obtain multi-scale representation. The cascaded structure presents more robust and meaningful features (as can be shown in Figure 1) for the FSL image recognition task. (3) Furthermore, a multi-label few-shot classification task is first proposed in this paper, which shows the effectiveness of the proposed FSL method from a more realistic and complex sample space. A new split of COCO data labeled as FS-COCO is compiled to benchmark the challenging yet important FSL task. (4) We also evaluate the cascaded architecture model on Omniglot and miniImageNet with our model demonstrating state-of-the-art results. (5) Finally, we construct four hard settings of Omniglot to evaluate the model's robustness in terms of size, location, and rotation variations. The source code is available on the website[1)].

2 Related work

2.1 Deep learning for few-shot image recognition

The few-shot image recognition presents many problems, which subsequently attract research attention in recent years. A lot of deep learning techniques have been proposed on the same problem. For example, to increase memory capacity, some studies adopted neural Turing machines [11, 12] or long short-term memory (LSTM) [13,14]. Also, some studies use parameter adaptation. In model-agnostic meta-learning (MAML) [9], the parameters are explicitly trained to generalize well on new tasks using a small number of gradient steps with a small amount of training data. Ravi and Larochelle [15] propose an LSTM based meta-learner model to learn the exact optimization algorithm used to train another learner neural network classifier in the few-shot regime.

There are also some specialized neural networks for few-shot image recognition. For example, a matching network [8] learns an embedding function with a sample-wise attention kernel to predict the similarity.

1) https://github.com/hustvl/cfmn.

Compared to matching network, prototypical network [10] has a similar structure but employs Euclidean distance instead of cosine distance. TADAM [16] proposes a dynamic task conditioned feature extractor based on the prototypical network. Unlike simple metrics, the relation network [17] learns a deep non-linear distance metric for similarity comparison. Some methods learn to predict the parameters for new categories without additional training [16,18–20], learn as a regression problem [21], learn from unlabeled data [22] or weakly-labeled data [23]. The simple neural attentive learner (SNAIL) uses temporal convolutions and soft attention to combine with the context of support samples. The transductive propagation network (TPN) [24] performs transductive learning on the similarity graph.

Data augmentation using generative models is also an effective option for the FSL [25, 26]. First, the attributed-guided augmentation methods in feature space are used in attribute guided augmentation (AGA) [27] and feature transfer network (FATTEN) [28]. Then, Hariharan and Girshick [29] transfer the transformation from a pair of known samples to a sample from a novel class. Δ-encoder [30] has a similar target as [29], but is trained as a reconstruction task. Also, Ref. [31] is more straightforward, which generates samples by adding random noises to support features. The diversity transfer network (DTN) [32] transfers diversity information from known categories to novel samples. Some methods use extra information, such as a deformation sub-network [33] or a pre-trained saliency network [34].

Therefore, our paper is a specialized neural network that can establish semantic associations between images and encourage the model to focus more on the features that have high correlations. Also, our proposed method overcomes the variance of inter-class and performs better when applied to few-shot image recognition.

2.2 Matching and attention for few-shot image recognition

Matching is an effective way to establish a semantic relationship between images [35–37], whereas the attention mechanism helps to decide which features are more useful based on the established relationships [5,38,39]. Matching network [8] uses a softmax function over the cosine distance between embedding features, as a sample-wise attention kernel. It treats each image as an individual sample without differentiating the semantic meanings of different pixels. In our paper, the attention is feature-wise between the query with each support image. It can learn the semantic association between each feature pair in different positions.

Moreover, attention is also applied between label semantics and image domains [40,41] for the few-shot image recognition, but they need extra information for word embedding. Our method learns from the training images only, without any other external information. Therefore, the attention mechanism we used is similar to self-attention [42, 43], which has proven to be effective on machine translation [44], image transformer [45], video sequence [46], and the generative adversarial network (GAN) [47]. Self-attention aims to find the relations within an image/sequence. Still, our method focuses more on establishing the correlation responses of each feature position between images for a more accurate similarity measure, which is especially considered for the few-shot image recognition. The STANet [48] is also similar to our method. But we combine the attention results from different feature expression levels, while the STANet only uses the high-level feature. The deep comparison network (DCN) [49] is also based on the Siamese structure to learn the relation between the query and support image. A sequence of relation modules is used to compute a non-linear metric. Nevertheless, our cascaded matching block focuses on matching fine-grained similarity of two compared images and highlights the matching feature to avoid interference from intra-class variance.

3 Method

3.1 Problem definition

To illustrate the few-shot image recognition task, we follow the definition in [8], which is defined as N-way K-shot learning. Each evaluation step is an N-way K-shot task that consists of two parts: support set, and query. We first sampled N classes from the training/testing set, then sampled a support set $\mathcal{D}_s = \{(x_s^i, y_s^i), i \in [1, \ldots, N \times K]\}$, which contains K labeled examples from each of the N classes. The query image (x_q, y_q) is sampled from the rest of the images from the N classes, i.e., $y_q \in \{y_s^i, i \in [1, \ldots, N \times K]\}$ and $x_q \notin \{x_s^i, i \in [1, \ldots, N \times K]\}$. It needs to be classified into one of the N classes based on the support set only. Different from traditional image recognition tasks based on lots of training images, the label

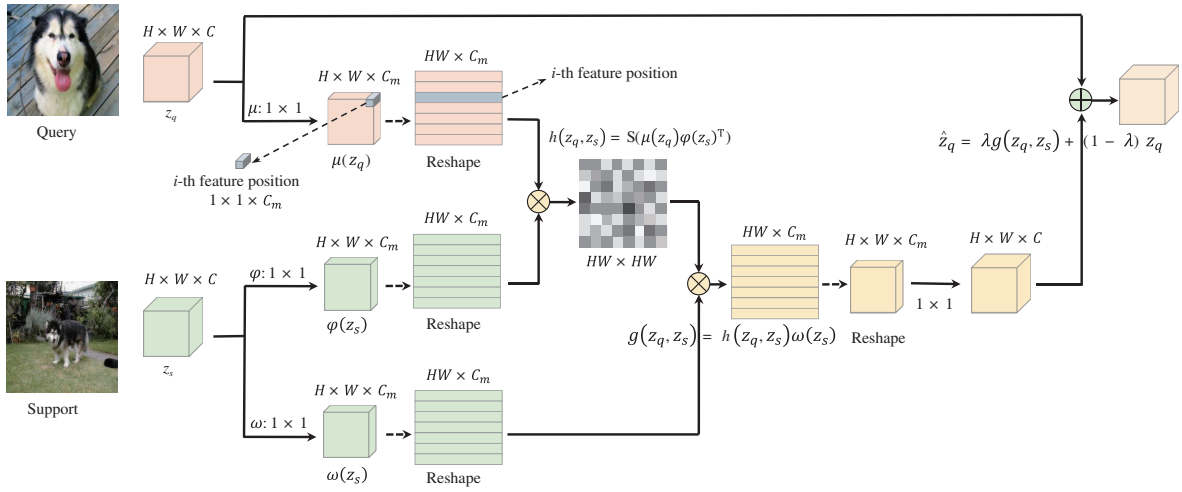

Figure 2 (Color online) Feature matching block. z_q and z_s are the features of the query and support image, respectively, which have the same shape $H \times W \times C$. After the space transformation μ, φ and the reshape operation, $h(z_q, z_s) = S(\mu(z_q)\varphi(z_s)^{\mathrm{T}})$ is a spatial attention map between each feature position of the query and it of the support image. S is the row-wise softmax. The feature $\omega(z_s)$ is scaled by the spatial attention map and mapped back to the input space. The final output of the block is the combination of the matched feature $g(z_q, z_s)$ and the original query feature z_q with the proportion of $\lambda : (1 - \lambda)$.

space of the training set here is disjointed with that of the testing set. The testing process is in the form of N-way K-shot, but with classes unknown to the training set.

3.2 Feature matching block

Figure 2 shows the details of the feature matching. The z_q and z_s are the features of the query and a support image from one of the hidden layers, respectively, which are both in the shape of $H \times W \times C$. First, they are mapped into another space μ and φ to get $\mu(z_q)$ and $\varphi(z_s)$, respectively. Next, they are reshaped to 2-dimensional matrices with the shape of $HW \times C_m$. The two matrices calculate a spatial attention map as follows:

$$h(z_q, z_s) = S(\mu(z_q)\varphi(z_s)^{\mathrm{T}}), \tag{1}$$

where S is the row-wise softmax. In the 3-dimensional metrics $\mu(z_q)$ and $\varphi(z_s)$, each feature point in $H \times W$ dimension is a feature position with the shape of $1 \times 1 \times C_m$, represented by $\mu(z_q^i)$ and $\varphi(z_s^i)$, $i \in [1, 2, \ldots, H \times W]$. After the reshaping, each row of the 2-dimensional matrix is a feature position, which is shown in In Figure 2. Therefore, each element $h^{i,j}$ of the spatial attention map is the similarity between the feature in the i-th position of the query and the feature in the j-th position of the support image as defined, as follows:

$$h^{i,j} = \frac{\exp(\mu(z_q^i)\varphi(z_s^j)^{\mathrm{T}})}{\sum_{j=1}^{H \times W} \exp(\mu(z_q^i)\varphi(z_s^j)^{\mathrm{T}})}. \tag{2}$$

Meanwhile, the support feature z_s is mapped to another space ω. It is scaled by the spatial attention map $h(z_q, z_s)$ to get $g(z_q, z_s) = h(z_q, z_s)\omega(z_s)$. Therefore, $\omega(z_s^j)$ indicates the feature in the j-th position of $\omega(z_s)$. Whereas a single feature position in $g(z_q, z_s)$ can be represented as follows:

$$g^i = \sum_{j=1}^{H \times W} h^{i,j}\omega(z_s^j). \tag{3}$$

We can find that the i-th feature position of the feature map $g(z_q, z_s)$ depends on the correlation responses between the i-th feature position of the query $\mu(z_q)$ with all the feature positions of the support $\varphi(z_s)$; hence, we describe it as a spatial attention mechanism. The features of z_q and z_s will be more retained if they are highly relevant to each other, while the irrelevant features tend to be ignored. Then, the network can learn to focus more on the relevant features, thereby reducing the influence of big variance and producing better results. Then matched feature $g(z_q, z_s)$ is mapped via a 1×1 convolution layer to get the same shape as the input z_q and z_s. Moreover, we find that keeping the original feature of the query image is helpful. Thus, to reach better similarity measurement in few-shot image recognition,

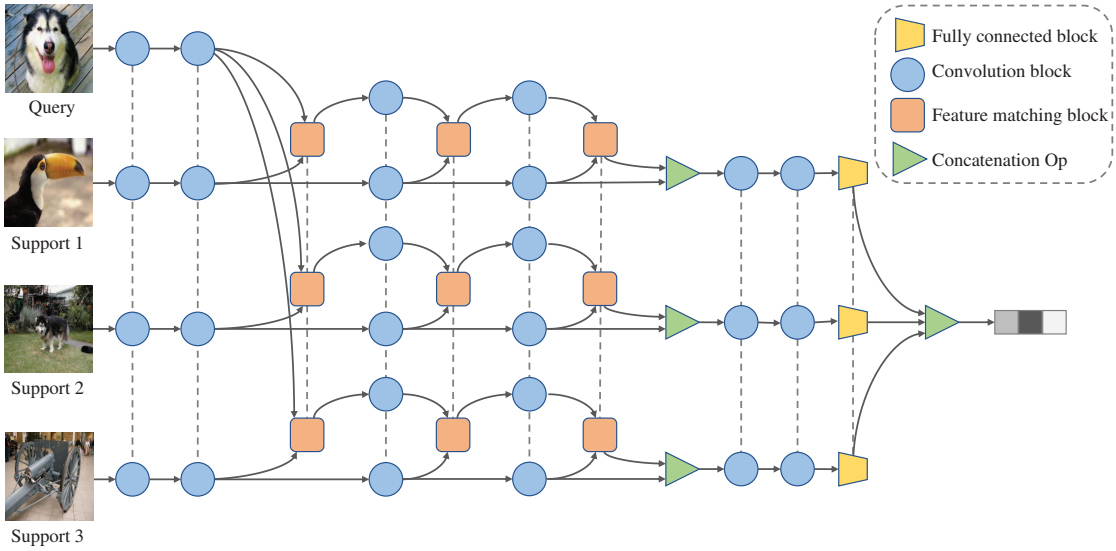

Figure 3 (Color online) Illustration of the proposed cascaded feature matching network. As shown in the top right corner of the figure, there are three different network blocks and one operation in CFMN. The blocks connected by a dashed line share the same parameters. Before the first concatenation operation, there are four convolutional blocks to extract the feature of each image. Three feature matching blocks are applied after the second, the third, and the fourth convolutional blocks, which form a cascaded structure. There are two convolutional blocks and one fully connected block to predict the similarity of the two concatenated features. The final prediction is the connection of all the similarity scores.

not only should the model focus on some particular parts that have high correlation responses, but it also takes the whole feature into account. So, the final output of the feature matching block is the combination of the matched feature $g(z_q, z_s)$ and the original query feature z_q with the proportion of $\lambda : (1-\lambda)$ is described as follows:

$$\hat{z}_q = \lambda g(z_q, z_s) + (1-\lambda) z_q, \quad \lambda \in [0, 1], \tag{4}$$

where λ is a weight factor over the matched feature. Notice, no matching information is injected if $\lambda = 0$; rather, the matched features are considered if $\lambda = 1$.

3.3 Cascaded feature matching network

In Figure 3, we take 3-way 1-shot for example. The overall structure is a conditional neural network $f(x_q, D_s; \theta)$ as we described in Subsection 3.1. The input consists of the query x_q (test image) and the support set D_s (condition). The output of the network is a 3-dimensional vector, which represents the prediction for x_q. The class with the highest prediction value is the final categorized result.

The first four convolutional blocks and all the three feature matching blocks can be viewed as a feature extractor. However, the extraction process of the query image depends on the feature matching results with corresponding support images. The cascaded structure combines matched information from different representation levels to achieve a more accurate and robust performance.

After the feature extraction process, the extracted features of the query and support images are concatenated in the channel dimension. Two convolution blocks and the fully connected block after the first concatenation operation learn a distance metric of the concatenated feature. The output of the fully connected block is a single value in a range of $[0, 1]$. The final output is the concatenation of all the three outputs of the fully connected block.

For K-shot where $K > 1$, the query will get K concatenated features with all K support images for one class. We perform an element-wise average over the K concatenated features to predict one similarity score for this class. Thus, it can be guaranteed that there are only N scores to form the final output.

3.4 CFMN for multi-label few-shot classification

We propose a multi-label extension to the traditional few-show classification problem, where each image may contain more than one interesting object. In this extended setting, the mapping between images and categories is many-to-many instead of many-to-one. As shown in Figure 4, taking 3-way 1-shot task, for

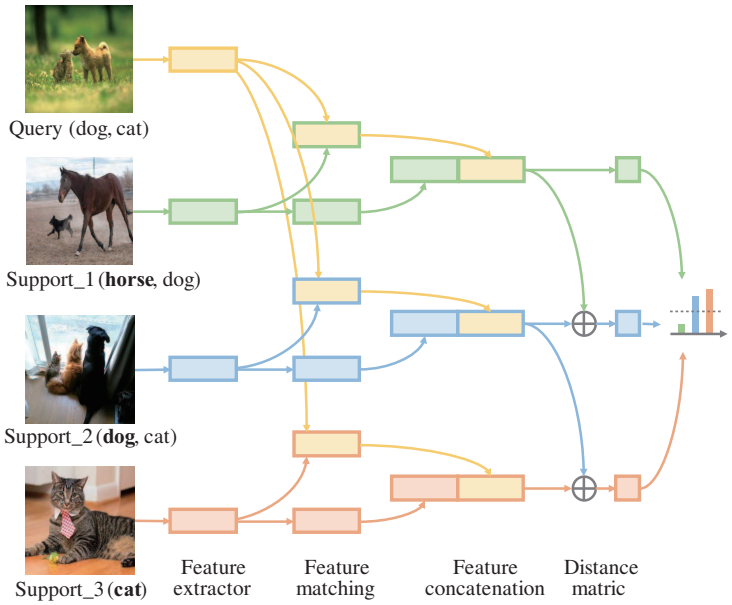

Figure 4 (Color online) Illustration of the CFMN for multi-label few-shot image classification. It shows a 3-label 3-way 1-shot task. The first support image is sampled as a horse image, but it also contains another interesting object, i.e., the dog. Therefore, while measuring the distance of the query and dog category, both the first and second support images are considered. The concatenated features of both of them are averaged before the distance metric procedure.

example, we first sampled 3 categories (horse, dog, cat) and sampled a support image for each category from all the images that contain the object. The first support image is sampled as a horse image, but it also contains another interesting object, i.e., the dog, and the query also belongs to more than one category. We believe that not only this setting brings up a more challenging and realistic problem to solve, but will also drive the model to learn a more generalized ability of images matching. Since the problem of memorization grows exponentially as the total number of categories, and the same image can become strong support but also a strong distractor under different queries. During inference, the final output is a 3-dimensional vector. The label values higher than a particular threshold (e.g., 0.4) are considered positive. In Section 4, we will show that our proposed method outperforms other previous methods in this problem.

4 Experiments

4.1 Dataset

Omniglot [50] was collected via Amazon Mechanical Turk to produce a standard benchmark for the FSL task of the handwritten character recognition domain. It contains 20 examples of 1623 characters from 50 different alphabets ranging from well-established international languages, which can be viewed as a transpose of the MNIST dataset. The images are resized to 28×28. Following [8,12], the data set is augmented with random rotations by multiples of 90 degrees. There are 1200 and 423 classes for training and testing, respectively.

miniImageNet was proposed in [8] by sampling a subset from the well-known ImageNet dataset [51]. It is a large-scale and challenging few-shot image classification dataset that consists of real-world images. It has served as a standard benchmark for many few-shot image classification methods. miniImageNet contains 100 classes, and each class has 600 images in the size of 84×84 pixels. Because the exact train-test splits used in [8] were not released, we followed the splits introduced by [15]. In this split setting, there are 64, 16, and 20 classes for training, validation, and testing, respectively.

FS-COCO is the first dataset for multi-label FSL proposed in this paper. It is a new split of the COCO dataset [52] that is one of the most popular datasets in multi-label classification. The COCO contains 80 classes in total. In our setting, the dataset is randomly divided into 54, 11, and 15 classes for training, validation, and testing, respectively. The details of the data split can be found in Appendix A. Since the ground-truth labels of the test set are not available, we only use the samples from the training

Table 1 The backbone of cascaded feature matching network for different datasets[a)]

Block name	miniImageNet & Omniglot		FS-COCO	
	Output size	Layers	Output size	Layers
CB 1	$41 \times 41 \times 64$	3×3 conv, 64 filters, BN, ReLU 2×2 maxpool, stride 2	$56 \times 56 \times 64$	7×7, 64, stride 2 3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
CB 2	$19 \times 19 \times 64$	3×3 conv, 64 filters, BN, ReLU 2×2 maxpool, stride 2	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
CB 3	$19 \times 19 \times 64$	3×3 conv, 64 filters, BN, ReLU	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
CB 4	$19 \times 19 \times 64$	3×3 conv, 64 filters, BN, ReLU	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
CO	$19 \times 19 \times 128$		$7 \times 7 \times 1024$	
CB 5	$8 \times 8 \times 64$	3×3 conv, 64 filters, BN, ReLU 2×2 maxpool, stride 2	$7 \times 7 \times 256$	3×3, 256, stride 1
CB 6	$3 \times 3 \times 64$	3×3 conv, 64 filters, BN, ReLU 2×2 maxpool, stride 2	$3 \times 3 \times 64$	3×3, 64, stride 1 2×2 max pool, stride 2
FCB	1	576×8 FC, ReLU 8×1 FC, Sigmoid	1	576×8 FC 8×1 FC, Sigmoid

a) CB: convolution block; CO: concatenation op; FCB: fully connected block. The left output size is calculated based on miniImageNet (84×84) for example.

set and validation set of version 2014 of COCO. The images are resized to 224×224 dimension.

4.2 Architecture

Most of the FSL models utilize four convolution layers for embedding feature extractor [8,9,17]. For a fair comparison, we follow the same architecture for miniImageNet and Omniglot, which is shown in Table 1. Here, each convolution block contains a 3×3 convolution layer followed by batch normalization and a ReLU non-linearity layer. The third and the fourth convolution blocks do not include the 2×2 max-pooling layer for providing a larger feature map to the distance metric network next to the convolution block. The concatenation operation is applied on the channel dimension. After the convolution block 6, the feature is reshaped to a vector and fed to the following two fully connected blocks. The final output is a single value that represents the similarity of the compared images. For the multi-label few-shot classification task on FS-COCO, a structure similar to ResNet-18 [1] is used, which is also shown in Table 1. The input size is 224×224 dimension.

4.3 Training details

We carry out 5-way 1-shot and 5-way 5-shot image classification experiments using the FS-COCO dataset. For each episode for 5-way 1-shot task, the support set is composed by sampling 1 image from each of the 5 classes. Then, we sampled another 15 samples, the query set from each of the 5 class in the remaining images for 1-shot task; thus, there are $1 \times 5 + 15 \times 5 = 80$ images in an episode/mini-batch for training. As for 5-way 5-shot classification, there are 5 images for each class in the support and query set, accordingly. Following [10], the model is trained on 20-way and 30-way 15 queries per training episode for miniImageNet. Besides, 5-way 1-shot and 5-way 5-shot, 20-way for 1-shot and 5-shot image classification experiments are also evaluated on Omniglot. There are 19 and 15 images for each class in the query set for 1-shot and 5-shot, respectively.

Our few-shot image classification network is trained on the training set and confirmed on the validation set. We selected the model that achieved the best performance on the validation set. The chosen model is evaluated on the testing set to obtain the final results. The mean square error loss is used to train our model.

We implemented the proposed network using PyTorch [53] and Adam optimizer [54]. The learning rate decreases by 0.1 to the initial one if the validation accuracy does not increase during the last 15000

Table 2 Few-shot images classification accuracies on Omniglot[a)]

Method	Reference	5-way 1-shot (%)	5-way 5-shot (%)	20-way 1-shot (%)	20-way 5-shot (%)
MANN [12]	ICML'16	82.8	94.9	–	–
Matching network [8]	NIPS'16	98.1	98.9	93.8	98.5
Neural statistician [55]	ICLR'17	98.1	99.5	93.2	98.1
ConvNet with memory module [56]	ICLR'17	98.4	99.6	95.0	98.6
Meta network [14]	ICML'17	99.0	–	97.0	–
Prototypical network [10]	NIPS'17	98.8	99.7	96.0	98.9
MAML [9]	ICML'17	98.7 ± 0.4	$\mathbf{99.9} \pm 0.1$	95.8 ± 0.3	98.9 ± 0.2
Relation network [17]	CVPR'18	99.6 ± 0.2	$\mathbf{99.8 \pm 0.1}$	97.6 ± 0.2	99.1 ± 0.1
CFMN (ours)	SCIS'20	$\mathbf{99.7 \pm 0.2}$	$\mathbf{99.8 \pm 0.1}$	$\mathbf{98.0 \pm 0.2}$	$\mathbf{99.2 \pm 0.1}$

a) '–': not reported. The best results are bold. The CFMN obtains state-of-the-art or comparable performance in all settings. Some accuracy results are reported with 95% confidence intervals.

Table 3 Few-shot images classification accuracies on miniImageNet[a)]

Method	Reference	5-way 1-shot (%)	5-way 5-shot (%)
Matching network [8]	NIPS'16	43.56 ± 0.84	55.31 ± 0.73
Meta network [14]	ICML'17	49.21 ± 0.96	–
Meta-learn LSTM [15]	ICLR'17	43.44 ± 0.77	60.60 ± 0.71
MAML [9]	ICML'17	48.70 ± 1.84	63.11 ± 0.92
Prototypical network [10]	NIPS'17	49.42 ± 0.78	68.20 ± 0.66
Relation network [17]	CVPR'18	50.44 ± 0.82	65.32 ± 0.70
CFMN (ours)	SCIS'20	$\mathbf{52.98 \pm 0.84}$	$\mathbf{68.33 \pm 0.70}$

a) '–': not reported. The best results are bold. The CFMN obtains the state-of-the-art performance on 5-way 1-shot and competitive results on 5-way 5-shot. All accuracy results are reported with 95% confidence intervals.

episodes. Besides, the current best model will be reloaded and trained with the updated learning rate. The training process is terminated if the validation accuracy does not increase during the last 50000 episodes.

4.4 Testing details

In testing and validation, there are 600 episodes for both MS-COCO and miniImageNet datasets. In every episode, 1 and 5 support images per class are sampled for the 1-shot setting and the 5-shot setting, respectively. Then, 15 images for each class are taken as the queries. Thus, we have $45000 = 600 \times 15 \times 5$ classification results. The mean and confidence intervals of the classification accuracy of the 45000 testing episodes are recorded. For the Omniglot dataset, there are 1000 testing episodes. In every episode, 19 and 15 query images per class are sampled for the 1-shot and the 5-shot setting, respectively.

To avoid the randomness of the episode sampling effects, we performed the above testing procedure 10 times. We recorded the mean of the accuracy and confidence intervals over all the 10 times in this paper.

4.5 Results

Results on Omniglot and miniImageNet. Tables 2 [8–10, 12, 14, 17, 55, 56] and 3 [8–10, 14, 15, 17] illustrate the performance of our proposed method against the current state-of-the-art on Omniglot and miniImageNet, accordingly. All accuracy results are reported with 95% confidence intervals. The best performing results are bold. It is noticed that our CFMN obtains better performance on both the two standard benchmarks than the state-of-the-art models, such as relation network, MAML, prototypical network, and meta network.

Multi-label few-shot learning results on FS-COCO. As shown in Table 4 [10, 17], precision, recall, and F1-measure are deployed to evaluate the models. The labels with confidence higher than 0.4 are considered positive. These measures do not require a fixed number of labels per image. Our model outperforms the existing methods significantly.

Impact of weight factor of matched feature. As defined in Subsection 3.2, λ represents the ratio of the matched feature and the original feature. $\lambda = 0.0$ depicts only using the original feature,

Table 4 Multi-label few-shot images classification accuracies on FS-COCO[a)]

Model	5-way 1-shot			5-way 5-shot		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
Prototypical network [10]	32.78	45.96	38.06	44.42	61.10	51.22
Relation network [17]	34.37	47.21	39.52	43.61	63.34	51.43
CFMN (ours)	**37.61**	**53.90**	**44.14**	**45.71**	**64.46**	**53.25**

a) The best results are bold. CFMN obtains the best performance.

Table 5 Impact of weight factor of matched feature[a)]

Weight factor	Accuracy (%)
CFMN with $\lambda = 0.00$	50.89
CFMN with $\lambda = 0.25$	52.02
CFMN with $\lambda = 0.50$	**52.98**
CFMN with $\lambda = 0.75$	50.28
CFMN with $\lambda = 1.00$	45.59

a) All the results are evaluated on miniImageNet for 5-way 1-shot task. The best results are bold.

Table 6 Impact of the details of the feature matching block[a)]

Model	Accuracy (%)	Model	Accuracy (%)
$C_m = 4$	51.63	$C_m = 64$	52.98
$C_m = 8$	52.14	$C_m = 128$	52.49
$C_m = 16$	52.52	w/o softmax	49.93
$C_m = 32$	52.93	w/o transformation	52.46

a) All the results are evaluated on miniImageNet for 5-way 1-shot task.

Table 7 Impact of the cascaded structure[a)]

Layers	Accuracy (%)	Layers	Accuracy (%)
CB 1	50.47	CB 3, 4	52.34
CB 2	51.11	CB 2, 3, 4	**52.98**
CB 3	51.63	CB 1, 2, 3, 4	50.17
CB 4	51.92		

a) All the results are evaluated on miniImageNet for 5-way 1-shot task. The best results are bold.

whereas $\lambda = 1.0$ indicates that the matched feature is taken into account only. We evaluated our model with several standard values for λ. From the results shown in Table 5, it can be found that the model cannot achieve the best performance with only the original feature alone or the matched feature alone. When $\lambda = 1$, the network only considers the matched information. But shallow layers only get some low-level vision information like the color, shape, and edge. Although feature matching is beneficial, an appropriate combination of the matched feature and the original feature is necessary. Making an analogy with how human beings recognize the similarity of two images, we would not only compare the details of them but also conclude by the visual context of the whole image. The combination by the ratio of λ behaves in the same way.

Impact of details of the feature matching block. Table 6 shows the impact of the reduction dim C_m, softmax axis, and space transformation operation. It can be seen that the accuracy does not just simply improve, as C_m increases. An appropriate setting for C_m achieves better performance and, at the same time, reduces the computation. The row-wise softmax and space transformation, both directly improve accuracy. Though, the row-wise softmax is more important to the results.

Impact of the cascaded structure. As defined in Subsection 3.3, we take a cascaded structure for combining the matched information from different representation levels to reach a more accurate and robust performance. To demonstrate the importance and effectiveness of the structure, we applied different numbers of feature matching blocks in various positions at the backbone. For example, convolution block $1, 2, 3, 4$ depicts that there are four feature matching blocks after the first fourth convolution blocks. From the results in Table 7, we can see that if taking only one feature matching block, deeper layers are

Figure 5 Samples of four harder variations on Omniglot. Original: Image size is 28×28. The characters are always in the center. Location: Images of the original set are randomly put in a 56×56 white background. Size: Characters are randomly resized to $[20, 55]$, and put in the center of the 56×56 white background. Rotation: Characters are resized to 50, and randomly rotated $[-45, 45]$ degrees, and put in the center of the 56×56 white background. All: Characters are randomly resized to $[20, 55]$, and randomly rotated $[-45, 45]$ degrees, and randomly put in the 56×56 white background.

Table 8 Results of four harder settings on Omniglot on 10-way 1-shot task[a)]

Weight factor	Original (%)	Size (%)	Location (%)	Rotation (%)	All (%)
Prototypical network [10]	98.02	95.75	94.34	93.67	88.93
Relation network [17]	99.18	98.95	97.64	96.94	94.95
CFMN (ours)	**99.23**	**98.99**	**99.05**	**98.42**	**97.89**

a) The best results are bold. Our CFMN always reaches the best performances. It can greatly reduce the influence caused by the differences in the size, location, rotation and even the combination of them.

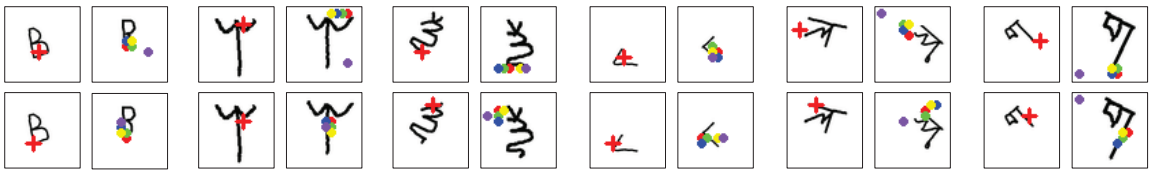

Figure 6 (Color online) Visualization of feature matching on the all-variation of Omniglot defined in Subsection 4.6. Two adjacent images form a group. The left one is the query. The red cross in it is an image position, which is matched with all the positions of the right support image. The colors, in turn, red, green, blue, yellow, and purple point to the positions which have the top five highest correlation responses.

better than the shallow one. Table 7 also shows that the cascaded structure is much better than only a single feature matching block. However, the exception is that the feature after the first convolution block is unsuitable for the matching block because the feature contains pixel information. Applying feature matching block here will make the model focus too much on the low-level feature, which has adverse effects on the performance.

4.6 How does CFMN work

The effectiveness of spatial feature matching. To further check the effectiveness of the feature matching block, we designed four harder variations (since query and support images are highly variant in location-variation, size-variation, rotation-variation, and all-variation). As shown in Figure 5, the image size of all the four harder variations is 56×56. Each image in the Omniglot is used to create ten different images. In the location-variation, we randomly placed the handwritten character on a white background. For the size-variation, we resized each character randomly, using $[20, 55]$ size range and put it in the center of a white background. Analogously, each image was randomly rotated by -45 to 45 degrees for the rotation-variation. The rotated images were also put in the center of a white background. As for all-variation, it combines all of the former operations for each image, which is more complex.

We evaluated our proposed CFMN on all the four harder variations and compared it with two existing methods. Table 8 [10, 17] shows that the CFMN consistently outperforms other methods, especially on the all-variation. The results on original Omniglot data are similar to each other. But the performances of the matching network and prototypical network decrease when dealing with harder visual differences. Therefore, our proposed model can overcome the obstacles from the object variations in terms of size, rotation, location, and even the combination of them.

Visualization. To provide a more intuitive view of how our proposed method works, we visualize the feature matching operation in Figures 6–8. Two images from the same class form a group in Figures 6 and 7. The left is the query, while the right is the supported image. The visualization is based on the spatial attention map in the last feature matching block. It stands for the performance of all three matching blocks because a matched feature is also the input of the next matching block. The feature has been matched three times after all the three matching blocks. The position represented by a red cross in the

Figure 7 (Color online) Visualization of feature matching on miniImageNet. The meaning of the red cross and colored dot is the same as Figure 6. Although the interested objects of each class may be different in terms of size, location, and style; they are associated together by our matching operation.

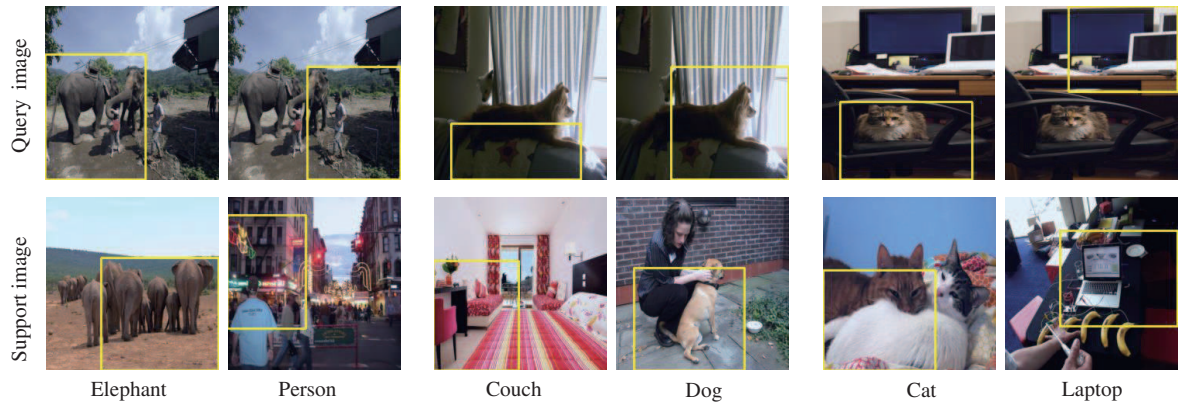

Figure 8 (Color online) Visualization of feature matching on FS-COCO. The yellow rectangular boxes indicate the receptive fields of the features that get the highest correlation responses in the last feature matching block. Two images aligned vertically is a group.

query is matched with all the right positions. By comparing the values in the spatial attention map, we point to positions with the top five highest correlation responses in different colors. It can be seen from figures that, although the compared characters are different in terms of size, location, and rotation, the corresponding strokes are associated together by our matching operation.

Since a deeper network is used for the FS-COCO, receptive fields of the features in the last feature matching block are more significant than those in miniImageNet and Omniglot. Therefore, the receptive field is depicted by the rectangular box in Figure 8. We can find that when the same query image is matched with different support images, the associated parts can get higher responses in the spatial attention map, which benefits a lot in the multi-label few-shot setting.

5 Conclusion

In this paper, we proposed a CFMN, which is a simple and effective method for few-shot image recognition. The fact that the interested object compared images from the real world usually differ significantly in terms of size, location, and style, spurs our motivation for this research. Our feature matching block can overcome those barriers and associate the corresponding parts together. The features with high correlation responses are paid more attention, whereas the opposite gets ignored naturally. Moreover, we apply three feature matching blocks to construct the cascaded structure that combines the matching information from various representation levels. The extensive experiments on a few-shot and multi-label few-shot classification using three standard datasets demonstrate the effectiveness of our proposed method. In the future, we hope to investigate more efficient feature matching using sparse self-attention for a few-shot image classification [57], few-shot object detection [58], and weakly-supervised object detection [59].

Acknowledgements This work was supported by National Natural Science Foundation of China (NSFC) (Grant Nos. 61876212, 61733007, 61572207) and HUST-Horizon Computer Vision Research Center.

References

1 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2016
2 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Neural Information Processing Systems (NeurIPS), 2012
3 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of International Conference on Learning Representations (ICLR), 2015
4 Wu Y H, Schuster M, Chen Z F, et al. Google's neural machine translation system: bridging the gap between human and machine translation. 2016. ArXiv:1609.08144
5 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2014. ArXiv:1409.0473
6 Oord A V D, Dieleman S, Zen H, et al. Wavenet: a generative model for raw audio. 2016. ArXiv:1609.03499
7 Bloom P. How Children Learn the Meanings of Words. Cambridge: MIT Press, 2000
8 Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning. In: Proceedings of Neural Information Processing Systems (NeurIPS), 2016
9 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of International Conference on Machine Learning (ICML), 2017
10 Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. In: Proceedings of Neural Information Processing Systems (NeurIPS), 2017
11 Graves A, Wayne G, Danihelka I. Neural Turing machines. 2014. ArXiv:1410.5401
12 Santoro A, Bartunov S, Botvinick M, et al. Meta-learning with memory-augmented neural networks. In: Proceedings of International Conference on Machine Learning (ICML), 2016
13 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780
14 Munkhdalai T, Yu H. Meta networks. In: Proceedings of International Conference on Machine Learning (ICML), 2017
15 Ravi S, Larochelle H. Optimization as a model for few-shot learning. In: Proceedings of International Conference on Learning Representations (ICLR), 2017
16 Oreshkin B, López P R, Lacoste A. TADAM: task dependent adaptive metric for improved few-shot learning. In: Proceedings of Neural Information Processing Systems (NIPS), 2018
17 Sung F, Yang Y X, Zhang L, et al. Learning to compare: relation network for few-shot learning. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
18 Qiao S Y, Liu C X, Shen W, et al. Few-shot image recognition by predicting parameters from activations. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
19 Qi H, Brown M, Lowe D G. Low-shot learning with imprinted weights. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
20 Gidaris S, Komodakis N. Dynamic few-shot visual learning without forgetting. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
21 Bertinetto L, Henriques J F, Torr P H, et al. Meta-learning with differentiable closed-form solvers. In: Proceedings of International Conference on Learning Representations (ICLR), 2019
22 Wang Y X, Hebert M. Learning to learn: model regression networks for easy small sample learning. In: Proceedings of European Conference on Computer Vision (ECCV), 2016
23 Liu L, Zhou T Y, Long G D, et al. Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph. In: Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI), 2019
24 Liu Y B, Lee J, Park M, et al. Learning to propagate labels: transductive propagation network for few-shot learning. In: Proceedings of International Conference on Learning Representations (ICLR), 2019
25 Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of International Conference on Computer Vision (ICCV), 2017
26 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of Neural Information Processing Systems (NIPS), 2014
27 Dixit M, Kwitt R, Niethammer M, et al. AGA: Attribute-guided augmentation. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2017
28 Liu B, Wang X D, Dixit M, et al. Feature space transfer for data augmentation. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
29 Hariharan B, Girshick R. Low-shot visual recognition by shrinking and hallucinating features. In: Proceedings of International Conference on Computer Vision (ICCV), 2017
30 Schwartz E, Karlinsky L, Shtok J, et al. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In: Proceedings of Neural Information Processing Systems (NIPS), 2018
31 Wang Y X, Girshick R, Hebert M, et al. Low-shot learning from imaginary data. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
32 Chen M T, Fang Y X, Wang X G, et al. Diversity transfer network for few-shot learning. In: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI), 2020
33 Chen Z T, Fu Y W, Wang Y X, et al. Image deformation meta-networks for one-shot learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019
34 Zhang H G, Zhang J, Koniusz P. Few-shot learning via saliency-guided hallucination of samples. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019
35 Thewlis J, Zheng S, Torr P H, et al. Fully-trainable deep matching. In: Proceedings of British Machine Vision Conference (BMVC), 2016
36 Novotnỳ D, Larlus D, Vedaldi A. Anchornet: a weakly supervised network to learn geometry-sensitive features for semantic matching. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2017
37 Wang Q Q, Zhou X W, Daniilidis K. Multi-image semantic matching by mining consistent features. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018
38 Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of International Conference on Machine Learning (ICML), 2015
39 Yang Z C, He X D, Gao J F, et al. Stacked attention networks for image question answering. In: Proceedings of Computer

Vision and Pattern Recognition (CVPR), 2016
40 Wang P, Liu L Q, Shen C H, et al. Multi-attention network for one shot learning. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2017
41 Chu W H, Wang Y C F. Learning semantics-guided visual attention for few-shot image classification. In: Proceedings of International Conference on Image Processing (ICIP), 2018
42 Cheng J P, Dong L, Lapata M. Long short-term memory-networks for machine reading. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2016
43 Parikh A P, Täckström O, Das D, et al. A decomposable attention model for natural language inference. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2016
44 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Neural Information Processing Systems (NeurIPS), 2017
45 Parmar N, Vaswani A, Uszkoreit J, et al. Image transformer. 2018. ArXiv:1802.05751
46 Wang X L, Girshick R, Gupta A, et al. Non-local neural networks. 2017. ArXiv:1711.07971
47 Zhang H, Goodfellow I, Metaxas D, et al. Self-attention generative adversarial networks. 2018. ArXiv:1805.08318
48 Yan S P, Zhang S Y, He X M, et al. A dual attention network with semantic embedding for few-shot learning. In: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI), 2019
49 Zhang X T, Sung F, Qiang Y T, et al. Deep comparison: relation columns for few-shot learning. 2018. ArXiv:1811.07100
50 Lake B M, Salakhutdinov R, Tenenbaum J B. Human-level concept learning through probabilistic program induction. Science, 2015, 350: 1332–1338
51 Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis, 2015, 115: 211–252
52 Lin T Y, Maire M, Belongie S, et al. Microsoft coco: common objects in context. In: Proceedings of European Conference on Computer Vision (ECCV), 2014
53 Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: Proceedings of Neural Information Processing Systems (NIPS) Workshop, 2017
54 Kinga D, Adam J B. A method for stochastic optimization. In: Proceedings of International Conference on Learning Representations (ICLR), 2015
55 Edwards H, Storkey A. Towards a neural statistician. In: Proceedings of International Conference on Learning Representations (ICLR), 2017
56 Kaiser Ł, Nachum O, Roy A, et al. Learning to remember rare events. In: Proceedings of International Conference on Learning Representations (ICLR), 2017
57 Huang Z L, Wang X G, Huang L C, et al. CCNET: criss-cross attention for semantic segmentation. In: Proceedings of International Conference on Computer Vision (ICCV), 2019. 603–612
58 Kang B Y, Liu Z, Wang X, et al. Few-shot object detection via feature reweighting. In: Proceedings of International Conference on Computer Vision (ICCV), 2019. 8420–8429
59 Tang P, Wang X G, Bai S, et al. PCL: proposal cluster learning for weakly supervised object detection. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 176–191

Appendix A

Data split for FS-COCO

Training set: toilet, teddy bear, bicycle, skis, tennis racket, snowboard, carrot, zebra, keyboard, scissors, chair, couch, boat, sheep, donut, tv, backpack, bowl, microwave, bench, book, elephant, orange, tie, bird, knife, pizza, fork, hair drier, frisbee, bottle, bus, bear, toothbrush, spoon, giraffe, sink, cell phone, refrigerator, remote, surfboard, cow, dining table, hot dog, baseball bat, skateboard, banana, person, train, truck, parking meter, suitcase, cake, traffic light.

Validation set: sandwich, kite, cup, stop sign, toaster, dog, bed, vase, motorcycle, handbag, mouse.

Testing set: laptop, horse, umbrella, apple, clock, car, broccoli, sports ball, cat, baseball glove, oven, potted plant, wine glass, airplane, fire hydrant.