

Learning to Focus: Cascaded Feature Matching Network for Few-shot Image Recognition

Mengting Chen¹, Xinggang Wang¹, Heng Luo², Yifeng Geng² & Wenyu Liu^{1*}

¹*Huazhong University of Science and Technology, Wuhan 430074, China;*

²*Horizon Robotics, Beijing 100080, China*

Abstract Deep networks can learn to accurately recognize objects of a category by training on a large number of annotated images. However, a meta-learning challenge known as a low-shot image recognition task comes when only a few images with annotations are available for learning a recognition model for one category. The objects in testing/query and training/support images are likely to be different in size, location, style, and so on. Our method, called Cascaded Feature Matching Network (CFMN), is proposed to solve this problem. We train the meta-learner to learn a more fine-grained and adaptive deep distance metric by focusing more on the features that have high correlations between compared images by the feature matching block which can align associated features together and naturally ignore those non-discriminative features. By applying the proposed feature matching block in different layers of the few-shot recognition network, multi-scale information among the compared images can be incorporated into the final cascaded matching feature, which boosts the recognition performance further and generalizes better by learning on relationships. The experiments for few-shot learning on two standard datasets, *miniImageNet* and *Omniglot*, have confirmed the effectiveness of our method. Besides, the multi-label few-shot task is first studied on a new data split of COCO which further shows the superiority of the proposed feature matching network when performing few-shot learning in complex images. The code will be made publicly available.

Keywords Few-shot learning, image recognition, feature matching

Citation Mengting Chen, Xinggang Wang, Heng Luo, et al. Title for citation. *Sci China Inf Sci*, for review

1 Introduction

Deep learning achieves great success in a variety of tasks with large amounts of labeled data for image recognition [14, 18, 38], machine translation [1, 48] and speech synthesis [26]. However, labeled data is not always massively available when annotation cost is too expensive or time is not allowed. By contrast, the human can learn novel concepts with only a few examples in a short time [3].

Few-shot learning attempts to resolve this problem by training a model that classifies an unlabeled example based on a small labeled support set. Specifically, N -way K -shot learning is the task of classifying an example, termed as a query, into one of N classes, when only K samples per class are available as supervision; these $N \times K$ samples with labels are termed as a support set. During training, support images and some query images are sampled. The meta-learner needs to distinguish the category of query images using only the support images. Moreover, the categories of the training set disjoint with those of testing set and they are randomly sampled to prevent direct semantic relationship and visual similarities between. Referring to [43], the batch of the support set and queries is termed as an *episode*.

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

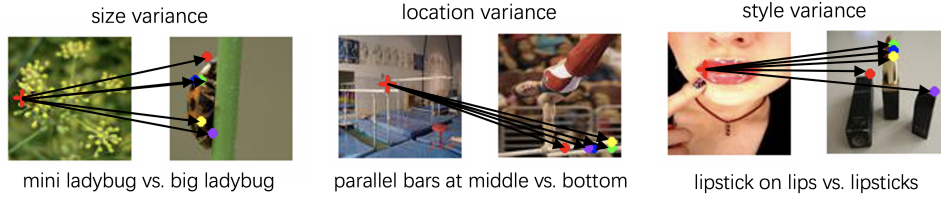


Figure 1 Visualization of the feature matching results of CFMN. Two adjacent images form a group. The feature at the red cross in the left query image matches with all features at the colored positions in the right support image. The colors, in turn, the red, green, blue, yellow, and purple point the positions which have the top five highest correlation responses. Although the interested objects may be different in size, location, style, and so on, they are associated together by our feature matching operation. More examples are shown in Fig. 7.

Given a test image, the model of few-shot learning needs to estimate the feature similarities between the test and the supporting images of each class. Different from the traditional image recognition task that 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. This means that the classifier needs to accurately evaluate the similarity with a little supervision and strong variance due to the lack of enough supporting information. As can be seen from Fig. 1, the query image may share limited visual similarities with support images. However, it's hard for the model to generalize among those strong inter-class differences with limited number of training images per class. [9, 39] show few-shot learning problem is prone to severe overfitting. To deal with the strong inter-class difference, we propose that the meta-learner should focus on essential spatial relationship features that have correlations between the query and support images and pay less attention to the non-discriminative features. We design our feature matching block to align the features of two compared images by the similarity of every feature position pairs. As shown in Fig. 1, two positions corresponding to the object from the same category will get a high response by our method, even the overall images may look quite different visually.

To fully utilize the proposed feature matching block, we apply three blocks at different layers of the network and cascade them together. The representation level of features from shallow layers of CNN is different from and usually lower than those of deep layers, and we extract the relation and similarity information of edges, shapes, and colors using shallow layers while 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, our main contributions are reflected in four aspects. **(1)** We propose a feature matching block that is capable of associating the object parts with high correlations between compared images and encouraging the model to pay more attention to those parts, which generalize to the large intra-class variation between the query and support images for few-shot learning challenge. **(2)** We cascade the feature matching block to obtain multi-scale representation. The cascaded structure obtains more robust and meaningful features (as can be shown in Fig. 1) for the few-shot image recognition task. **(3)** The multi-label few-shot classification task is first proposed in this paper which shows the effectiveness of the proposed method for few-shot learning from a more realistic and complex sample space. A new split of COCO termed as FS-COCO, is compiled to benchmark this difficult yet important few-shot learning task. **(4)** We also evaluate the cascaded architecture model on Omniglot and *miniImageNet*. Our model shows state-of-the-art results. **(5)** We construct four hard settings of Omniglot to evaluate the model's robustness on size, location, and rotation variations.

2 Related work

2.1 Deep learning for few-shot image recognition

Few-shot image recognition is a challenging problem which gains increasing attention in recent years. A lot of deep learning techniques have sprung up. In order to increase memory capacity, some works

adopted Neural Turing Machines [12, 36] or LSTM [15, 24]. There are also some works using parameter adaptation. In MAML [9], the parameters are explicitly trained to generalize well on new tasks by a small number of gradient steps with a small amount of training data. Sachin and Hugo [34] 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. Matching Network [43] learns an embedding function with a sample-wise attention kernel to predict the similarity. Compared to Matching Network, Prototypical Network [39] has a similar structure but employs Euclidean distance instead of cosine distance. TADAM [28] proposes a dynamic task conditioned feature extractor based on Prototypical Network. Different from simple metrics, Relation Network [40] learns a deep non-linear distance metric for similarity comparing. There are also some methods learn to predict the parameters for novel categories without additional training [10, 27, 32, 33], learn as a regression problem [2], learn from unlabeled data [47] or weakly-labeled data [22]. SNAIL uses temporal convolutions and soft attention to combine with the context of support samples. TPN [23] performs transductive learning on the similarity graph. DTN [4] generates new reference features by transferring diversity information between training image pairs in the same class. [50] learns object parts by clustering cell features and modeling their relationships in an attentional manner for few shot learning, which obtains the state-of-the-art performance on the few-shot image classification benchmarks.

Data augmentation using generative models is also an effective option for few-shot learning [11, 56]. At first, attributed-guided augmentation methods in feature space are used in AGA [7] and FATTEN [21]. Then Hariharan and Girshick [13] transfer the transformation from a pair of known samples to a sample from a novel class. Δ -encoder [37] has similar target as [13], but it is trained as a reconstruction task. [45] is more straightforward which generates samples by adding random noises to support features. There are also some methods used extra information, such as a deformation sub-network [57] or a pre-trained saliency network [54].

Our work 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; it overcomes the variance of inter-class and gets better performance for few-shot image recognition.

2.2 Matching and attention for few-shot image recognition

Matching is an effective way to establish semantic correspondences between images [25, 41, 45]; and the attention mechanism can help to decide which features are more useful based on the established correspondences [1, 49, 52]. Matching Network [43] uses the 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 work, the attention is feature-wise between the query with each support images. It can learn the semantic correspondences between each feature pair in different positions.

Attention can also be applied between label semantics and image domains [6, 44] for 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. Our attention mechanism is similar to self-attention [5, 29] which has proven to be effective on machine translation [42], image transformer [30], video sequence [46] and GAN [53]. Self-attention aims to find the relations within an image/sequence, but our method focuses more on establishing the correlation responses of each feature position between images for more accurate similarity measure which is specially designed for few-shot image recognition. The STANet [51] is also similar to us. But we combine the attention results from different feature expression levels, while STANet only uses the high-level feature. DCN [55] 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. But our cascaded matching block focuses on matching fine-grained similarity of two compared images, and highlights the corresponding feature to avoid interference from intra-class variance.

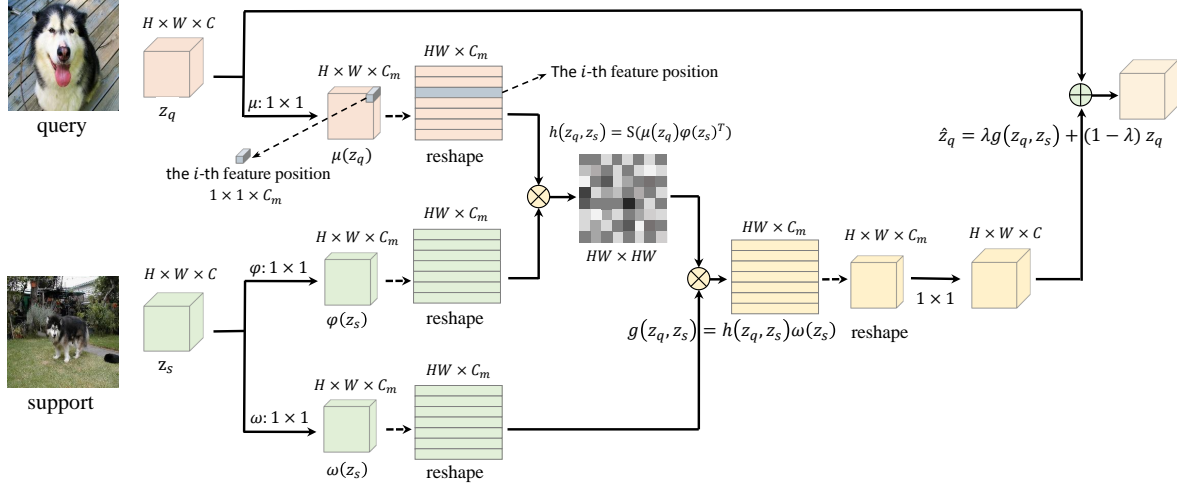


Figure 2 Feature matching block. z_q and z_s are the features of the query and support image, respectively, which have the same shape $H \times W \times C$. After the space transformation μ , φ and the reshape operation, $h(z_q, z_s) = S(\mu(z_q)\varphi(z_s)^T)$ is a spatial attention map between each 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 $\lambda : (1 - \lambda)$.

3 Method

3.1 Problem definition

To illustrate the few-shot image recognition task, we follow the definition in [43] which is termed as N -way K -shot learning. Each evaluation step is an N -way K -shot task which consists of two parts, support set, and query. We first sample N classes from the training/testing set, then sample a support set $\mathcal{D}_s = \{(x_s^i, y_s^i), i \in [1, \dots, N \times K]\}$, which contains K labeled examples from each of the N classes. The query image (x_q, y_q) is sampled from the rest images of the N classes, i.e. $y_q \in \{y_s^i, i \in [1, \dots, N \times K]\}$ and $x_q \notin \{x_s^i, i \in [1, \dots, N \times K]\}$. It needs to be classified into one of the N classes based only on the support set. Different from traditional image recognition tasks based on lots of training images, the label space of the training set here is disjointed with it 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

The details of the feature matching block are shown in Fig. 2. 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 $H \times W \times C$. Firstly, they are mapped into another space μ and φ to get $\mu(z_q)$ and $\varphi(z_s)$ respectively. Then 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,

$$h(z_q, z_s) = S(\mu(z_q)\varphi(z_s)^T), \quad (1)$$

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 $\varphi(z_s^i)$, $i \in [1, 2, \dots, H \times W]$. After reshaping, each row of the 2-dimensional matrix is a feature position which is shown in In Fig. 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,

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

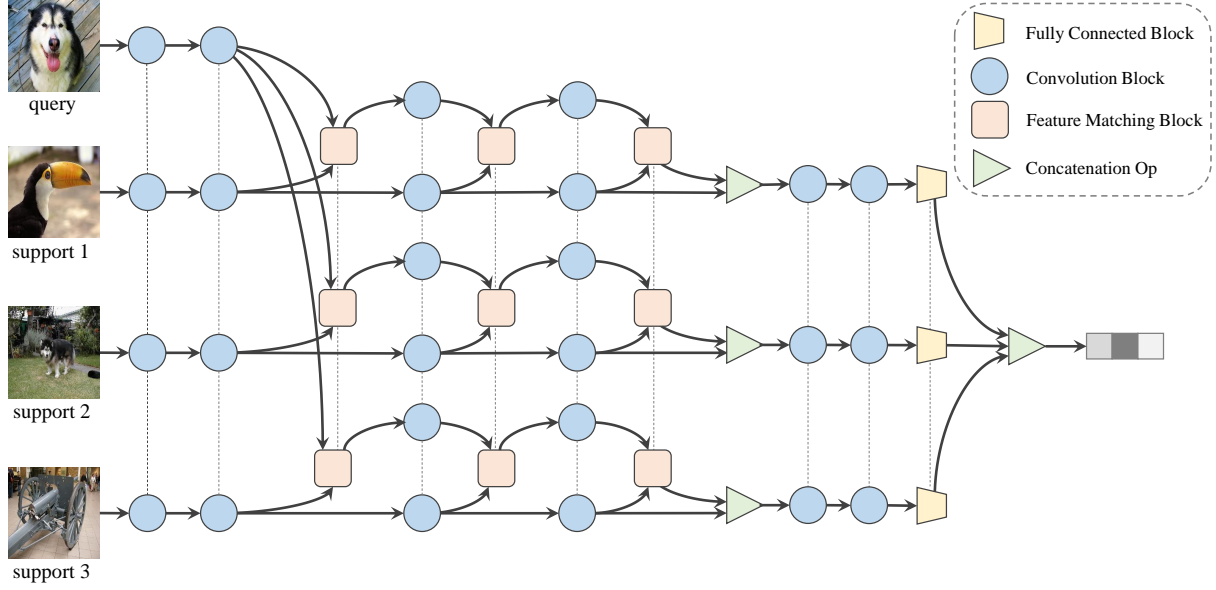


Figure 3 Illustration of the proposed Cascaded Feature Matching Network. As shown in the top right corner of the figure, there are three different network blocks and one operation in CFMN. The blocks connected by a dashed line share the same parameters. Before the first Concatenation Operation, there 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

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 $\omega(z_s)$. A single feature position in $g(z_q, z_s)$ can be represented as follows,

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

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 $\varphi(z_s)$. That is why we term 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 and the irrelevant features tend to be ignored. Then the network can learn to focus more on the relevant features, thereby reducing the influence of strong variance and producing better results. Then the 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. In few-shot image recognition, in order to reach better similarity measurement, not only should the model focus on some particular parts that have high correlation responses, but 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,

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

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

3.3 Cascaded Feature Matching Network

In Fig. 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 Sect. 3.1. The input consists of the query x_q (test image) and the support

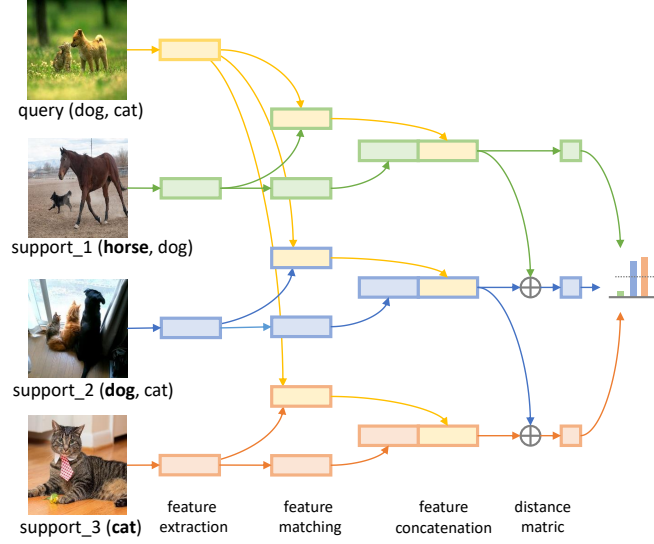


Figure 4 Illustration of CFMN for multi-label few-shot image classification. It shows a 3-label 3-way 1-shot task. The first support image is sampled as a horse image, but it also contains another interested object, *i.e.*, the dog. Therefore, during measuring the distance of the query and the dog category,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

set D_s (condition). The output of the network is a 3-dimensional vector which indicates 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 is dependent on the feature matching results with respective support images. The cascaded structure combines matched information from different representation levels to reach a more accurate and robust performance.

After the feature extraction process, 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 element-wise average over those 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 interested object. In this extended setting, the mapping between images and categories is many-to-many instead of many-to-one. As shown in Fig. 4, taking 3-way 1-shot task for example, we first sample 3 categories (horse, dog, cat) and sample 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 interested 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 difficult and realistic problem to solve, but will also drive the model to learn a more generalized ability of images matching. Since the difficulty 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 surpasses other previous methods in this problem.

Table 1 The backbone of Cascaded Feature Matching Network for different datasets. CB: Convolution Block; CO: Concatenation Op; FCB: Fully Connected Block. The left output size is calculated based on *miniImageNet* (84×84) for example.

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	$56 \times 56 \times 64$	$7 \times 7, 64$, 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	$28 \times 28 \times 128$	$3 \times 3, 128$ $3 \times 3, 128$ $\times 2$
				$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
CB 3	$19 \times 19 \times 64$	3×3 conv, 64 filters, BN, ReLU	$14 \times 14 \times 256$	$3 \times 3, 256$ $3 \times 3, 256$ $\times 2$
				$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
CB 4	$19 \times 19 \times 64$	3×3 conv, 64 filters, BN, ReLU	$7 \times 7 \times 512$	$3 \times 3, 512$ $3 \times 3, 512$ $\times 2$
				$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
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	$7 \times 7 \times 256$	$3 \times 3, 256$, stride 1
CB 6	$3 \times 3 \times 64$	3×3 conv, 64 filters, BN, ReLU 2×2 maxpool, stride 2	$3 \times 3 \times 64$	$3 \times 3, 64$, stride 1
				2×2 max pool, stride 2
FCB	1	576×8 FC, ReLU	1	576×8 FC
		8×1 FC, Sigmoid		8×1 FC, Sigmoid

4 Experiments

4.1 Dataset

Omniglot [19] was collected via Amazon’s Mechanical Turk to produce a standard benchmark for the few-shot learning 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 dataset MNIST. The images are resized to 28×28 . Following [36, 43], 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 [43] by sampling a subset from the well-known ImageNet dataset [35]. It is a large-scale and challenging few-shot image classification dataset that consists of real-world images, and it has been 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 [43] were not released, we followed the splits introduced by [34]. 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 few-shot learning proposed in this paper. It is a new split of the COCO dataset [20], which is one of the most popular datasets in multi-label classification. 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 the Appendix. Since the ground-truth labels of the test set are not available, we only use the samples from the training set and validation set of version 2014 of COCO. The images are resized to 224×224 .

4.2 Architecture

Most few-shot learning models utilize four convolution layers for embedding feature extractor [9, 40, 43]. For a fair comparison, we follow the same architecture for *miniImageNet* and Omniglot which is shown in Table 1. 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 contain the 2×2 max-pooling layer for providing a larger feature map to the following distance metric network. 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 represents the similarity of the compared images. For the multi-label few-shot classification task on FS-COCO, a structure similar to ResNet-18 [14] is used which is also shown in Table 1. The input size is 224×224 .

4.3 Training details

We carry out 5-way 1-shot and 5-way 5-shot image classification experiments for FS-COCO. For each episode on the 5-way 1-shot task, the support set is composed by sampling 1 image from each of the 5 classes; then we sample another 15 samples as the query set from each of the 5 class among the remain images for 1-shot task; thus there are $1 \times 5 + 15 \times 5 = 80$ images in a *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, respectively. Following [39], the model is trained on 20-way and 30-way 15 queries per training episode for *miniImageNet*. Beside 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 validated on the validation set. The model that obtains the best performance on the validation set is selected. The selected model is evaluated on the testing set to obtain the final results. The mean square error (MSE) loss is used to train our model.

We implement the proposed network using PyTorch [31]. The optimizer is Adam [17]; the learning rate decreases by 0.1 to the original one if the validation accuracy does not increase during the last 15,000 *episode*. Besides, the current best model will be reloaded and trained with the updated learning rate. The training procedure is early stopped if the validation accuracy does not increase during the last 50,000 *episode*.

4.4 Testing details

In testing and validation, there are 600 episodes for datasets MS-COCO and *miniImageNet*. 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 $45,000 = 600 \times 15 \times 5$ classification results. The mean and confidence intervals of the classification accuracy of the 45,000 testings are recorded. For dataset Omniglot, 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 perform the above testing procedure for 10 times. The mean of the accuracy and confidence intervals over all the 10 times are reported in this paper.

4.5 Results

Results on Omniglot and *miniImageNet*

Table 2 and 3 illustrate the performance of our method against the current state-of-the-art on Omniglot and *miniImageNet*, respectively. All accuracy results are reported with 95% confidence intervals. The best performing results are bold. It can be observed that our CFMN obtains better performance on both the two classical benchmarks than the state-of-the-art models, such as Relation Network, MAML, Prototypical Network, Meta Network.

Multi-label few-shot learning results on FS-COCO

As shown in Table 4, precision, recall, and F1-measure are employed to evaluate the models. 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 by a sizable margin.

Impact of weight factor of matched feature

Table 2 Few-shot images classification accuracies on Omniglot. ‘-’: not reported. The best results are bold. The Cascaded Feature Matching Network (CFMN) obtains the state-of-the-art or comparable performance on all settings. Some accuracy results are reported with 95% confidence intervals.

Methods	Ref	5-way 1-shot	5-way 5-shot	20-way 1-shot	20-way 5-shot
MANN [36]	ICML’16	82.8%	94.9%	-	-
Matching Network [43]	NIPS’16	98.1%	98.9%	93.8%	98.5%
Neural Statistician [8]	ICLR’17	98.1%	99.5%	93.2%	98.1%
ConvNet with Memory Module [16]	ICLR’17	98.4%	99.6%	95.0%	98.6%
Meta Network [24]	ICML’17	99.0%	-	97.0%	-
Prototypical Network [39]	NIPS’17	98.8%	99.7%	96.0%	98.9%
MAML [9]	ICML’17	98.7% \pm 0.4%	99.9% \pm 0.1%	95.8% \pm 0.3%	98.9% \pm 0.2%
Relation Network [40]	CVPR’18	99.6% \pm 0.2%	99.8% \pm 0.1%	97.6% \pm 0.2%	99.1% \pm 0.1%
CFMN (Ours)		99.7% \pm 0.2%	99.8% \pm 0.1%	98.0% \pm 0.2%	99.2% \pm 0.1%

Table 3 Few-shot images classification accuracies on miniImageNet. ‘-’: not reported. The best results are bold. The Cascaded Feature Matching Network (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.

Methods	Ref	5-way 1-shot	5-way 5-shot
Matching Network [43]	NIPS’16	43.56% \pm 0.84%	55.31% \pm 0.73%
Meta Network [24]	ICML’17	49.21% \pm 0.96%	-
Meta-Learn LSTM [34]	ICLR’17	43.44% \pm 0.77%	60.60% \pm 0.71%
MAML [9]	ICML’17	48.70% \pm 1.84%	63.11% \pm 0.92%
Prototypical Network [39]	NIPS’17	49.42% \pm 0.78%	68.20% \pm 0.66%
Relation Network [40]	CVPR’18	50.44% \pm 0.82%	65.32% \pm 0.70%
CFMN (Ours)		52.98% \pm 0.84%	68.33% \pm 0.70%

Table 4 Multi-label few-shot images classification accuracies on FS-COCO. The best results are bold. CFMN obtains the best performance.

Model	5-way 1-shot			5-way 5-shot		
	Precision	Recall	F1	Precision	Recall	F1
Prototypical Network [39]	32.78%	45.96%	38.06%	44.42%	61.10%	51.22%
Relation Network [40]	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%

As defined in Sect. 3.2, λ represents the ratio of the matched feature and the original feature. $\lambda = 0.0$ means only using the original feature while $\lambda = 1.0$ means that only the matched feature is taken into account. We evaluated our model with several standard values for λ . Referring to the results shown in Table 5, it can be found that the model cannot reach the best performance with whether the original feature alone or the matched feature alone. When $\lambda = 1$, the network only takes the matched information into consideration. But shallow layers only get some low-level vision information like the color, shape, and edge. Although feature matching is really helpful, an appropriate combination of the matched feature and the original feature is necessary. Making an analogy with how our 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 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 , the 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 can get better performance, at the same time reduce the computation. The row-wise softmax and space transformation both directly improve accuracy. But obviously, the row-wise softmax is more important to the results.

Impact of the cascaded structure

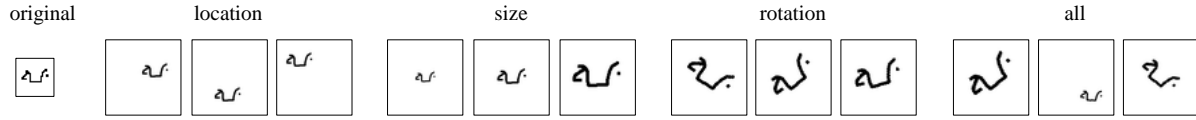
As defined in Sect. 3.3, we take a cascaded structure for combining the matched information from

Table 5 Impact of weight factor of matched feature. All the results are evaluated on *miniImageNet* for 5-way 1-shot task. The best results are bold.

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%

Table 6 Impact of the details of the feature matching block. All the results are evaluated on *miniImageNet* for 5-way 1-shot task.

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%

**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 7 Impact of the cascaded structure.** All the results are evaluated on *miniImageNet* for 5-way 1-shot task. The best results are bold.

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%		

different representation levels to reach a more accurate and robust performance. In order to illustrate the necessity and effectiveness of this structure, we applied the different numbers of feature matching blocks in different positions at the backbone. For example, Convolution Block 1, 2, 3, 4 means that there are four feature matching blocks after the first fourth Convolution Blocks, respectively. From the results in Table 7, we can see that if taking only one feature matching block, deeper layers are better than the shallow one. Table 7 also shows that the cascaded structure is much better than only a single one feature matching block. However, the exception is that the feature after the first Convolution Block is unsuitable for the matching block. Because the feature merely contains pixel information. Applying feature matching block here will make the model focus too much on the low-level feature which has detrimental effects on the performance.

4.6 How does CFMN work

The effectiveness of spatial feature matching

In order to further check the effectiveness of the feature matching block, we design four harder variations (query and support images are highly variant in *location*-variation, *size*-variation, *rotation*-variation and *all*-variation). As shown in Fig. 5, the image size of all the four harder variations is 56×56 . Each image in the Omniglot is used to create 10 different images. In the *location*-variation, we randomly place the

Table 8 Results of four harder settings on Omniglot on 10-way 1-shot task. 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.

weight factor	original	size	location	rotation	all
Prototypical Network [39]	98.02%	95.75%	94.34%	93.67%	88.93%
Relation Network [40]	99.18%	98.95%	97.64%	96.94%	94.95%
CFMN (Ours)	99.23%	98.99%	99.05%	98.42%	97.89%

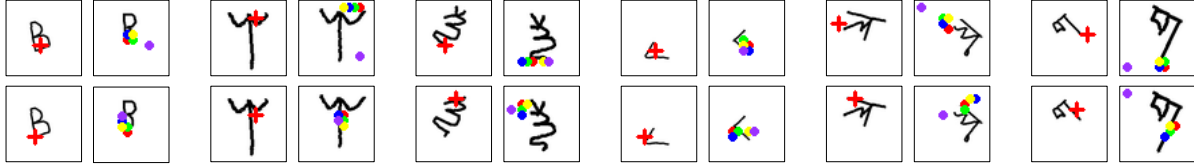


Figure 6 Visualization of feature matching on the *all*-variation of Omniglot defined in Sect 4.6. Two adjacent images form a group. The left one is the query. The red cross in it is an image position which is matched with all positions of the right support image. The colours, in turn, the red, green,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



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

handwritten character on a white background. For the *size*-variation, we randomly resize each character by the size range in [20, 55] and put it in the center of a white background. Analogously, each image is randomly rotated by -45 to 45 degrees for the *rotation*-variation. The rotated images are also put in the center of a white background. As for *all*-variation, as the name implies, it combines all of the former operations for each image, which is more difficult than the other.

We evaluate our CFMN on all the four harder variations and compare it with two existing methods. We can see from the results shown in Table 8 that CFMN consistently outperforms the other works, especially on the *all*-variation. The results on original Omniglot data are similar to each other. But the performances of Matching Network and Prototypical Network severely decrease when dealing with harder visual differences. It illustrates that our proposed model can overcome the obstacles from the object variations in the 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 Fig. 6, Fig. 7 and Fig. 8. Two images from the same class form a group in Fig. 6 and Fig. 7. The left is the query; the right is the support image. The visualization is based on the spatial attention map in the last feature matching block. It stands for the performance of all of the 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 of three matching blocks. The position represented by a red cross in the query is matched with all the right positions. By comparing the values in the spatial attention map, we point positions which have the top five highest correlation responses by different colors. It can be seen from figures that although the compared characters are different in the size, location, rotation, and so on, the corresponding strokes are associated together by our matching operation.

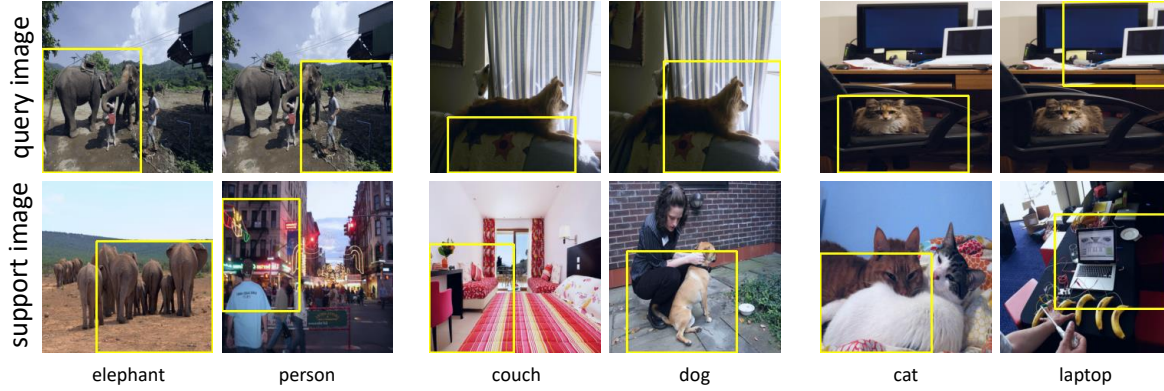


Figure 8 Visualization of feature matching on FS-COCO. The yellow rectangular boxes indicate the receptive fields of the features that get the highest correlation responses in the last Feature Matching Block. Two images aligned vertically is a group.

Since a deeper network is used for FS-COCO, receptive fields of the features in the last Feature Matching Block is larger than them in *miniImageNet* and *Omniglot*. Therefore, the receptive field is indicated by the rectangular box in Fig. 8. We can find that when the same query image 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 the Cascaded Feature Matching Network (CFMN), which is a simple and effective method for few-shot image recognition. Our motivation is based on the observation that the interested object in compared images from the real world usually differs significantly in the size, location, style, *etc.* Our feature matching block can overcome those barriers and associate the corresponding parts together. The features with high correlation responses are paid more attention, while the opposite will be naturally ignored. Three feature matching blocks are applied there to construct a cascaded structure that combines the matching information from different representation levels. The extensive experiments on few-shot and multi-label few-shot classification on three standard datasets demonstrate the effectiveness of our proposed method.

Appendix

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.

Acknowledgements This work was supported by NSFC (No. 61876212, No. 61733007 and No. 61572207) and HUST-Horizon Computer Vision Research Center.

References

- 1 Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

- 2 Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2019.
- 3 Paul Bloom. *How children learn the meanings of words*, volume 377. Citeseer, 2000.
- 4 Mengting Chen, Yuxin Fang, Xinggang Wang, Heng Luo, Yifeng Geng, Xinyu Zhang, Chang Huang, Wenyu Liu, and Bo Wang. Diversity transfer network for few-shot learning. In *AAAI*, pages 10559–10566, 2020.
- 5 Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- 6 Wen Hsuan Chu and Yu Chiang Frank Wang. Learning semantics-guided visual attention for few-shot image classification. In *International Conference on Image Processing (ICIP)*, 2018.
- 7 Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- 8 Harrison Edwards and Amos Storkey. Towards a neural statistician. *International Conference on Learning Representations (ICLR)*, 2017.
- 9 Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning (ICML)*, 2017.
- 10 Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- 11 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, 2014.
- 12 Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.
- 13 Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *International Conference on Computer Vision (ICCV)*, Oct 2017.
- 14 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- 15 Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- 16 Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *International Conference on Learning Representations (ICLR)*, 2017.
- 17 D Kinga and J Ba Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- 18 Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2012.
- 19 Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- 20 Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision (ECCV)*, 2014.
- 21 Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- 22 Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Prototype propagation networks (ppn) for weakly-supervised few-shot learning on category graph. 2019.
- 23 Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019.
- 24 Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International conference on machine learning (ICML)*, 2017.
- 25 David Novotný, Diane Larlus, and Andrea Vedaldi. AnchorNet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- 26 Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- 27 Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2018.
- 28 Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Neural Information Processing Systems (NIPS)*, 2018.
- 29 Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- 30 Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, and Alexander Ku. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.
- 31 Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- 32 Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- 33 Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

- 34 Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.
- 35 Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- 36 Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning (ICML)*, 2016.
- 37 Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Neural Information Processing Systems (NIPS)*. 2018.
- 38 Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.
- 39 Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2017.
- 40 Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- 41 James Thewlis, Shuai Zheng, Philip HS Torr, and Andrea Vedaldi. Fully-trainable deep matching. In *British Machine Vision Conference (BMVC)*, 2016.
- 42 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, 2017.
- 43 Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2016.
- 44 Peng Wang, Lingqiao Liu, Chunhua Shen, Zi Huang, Anton van den Hengel, and Heng Tao Shen. Multi-attention network for one shot learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- 45 Qianqian Wang, Xiaowei Zhou, and Kostas Daniilidis. Multi-image semantic matching by mining consistent features. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- 46 Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 2017.
- 47 Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European conference on computer vision (ECCV)*, 2016.
- 48 Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
- 49 Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning (ICML)*, 2015.
- 50 Weijian Xu, Yifan Xu, Huaijin Wang, and Zhuowen Tu. Constellation nets for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021.
- 51 Shipeng Yan, Songyang Zhang, Xuming He, et al. A dual attention network with semantic embedding for few-shot learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.
- 52 Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- 53 Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.
- 54 Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- 55 Xueting Zhang, Flood Sung, Yuting Qiang, Yongxin Yang, and Timothy M Hospedales. Deep comparison: Relation columns for few-shot learning. *arXiv preprint arXiv:1811.07100*, 2018.
- 56 Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017.
- 57 Yu-Xiong Wang Lin Ma Wei Liu Martial Hebert Zitian Chen, Yanwei Fu. Image deformation meta-networks for one-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.