Information Sciences 611 (2022) 464–475

Contents lists available at ScienceDirect

Information Sciences

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

Multi-instance attention network for few-shot learning

Zhili Qin [a,b], Han Wang [a,b], Cobbinah Bernard Mawuli [a,b], Wei Han [a], Rui Zhang [a], Qinli Yang [a,b], Junming Shao [a,b,*]

[a] Data Mining Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou313001, China

ARTICLE INFO

Article history:
Received 8 November 2021
Received in revised form 5 May 2022
Accepted 2 July 2022
Available online 12 July 2022

Keywords:
Few-shot learning
Multi-instance learning
Self-attention network

ABSTRACT

The attention mechanism is usually equipped with a few-shot learning framework and plays a key role in extracting the semantic object(s). However, most attention networks in existing few-shot learning algorithms often work on the channel and/or pixel dimension, leading to the size of attention maps being large. Due to lack of training examples, these attention networks are prone to over-fitting, and may fail to find the semantic target(s). In this paper, we split the original image into patches, extending a new dimension in image data, namely, the patch dimension. On the one hand, the number of patch dimensions is usually much smaller than the traditional three dimensions, thus greatly reducing the number of attention module parameters. On the other hand, the patch dimensional attention mechanism can benefit from multi-instance learning and achieve a good compromise between global and local features. Four comparison experiments on four typical real-world data sets (miniImageNet, tieredImageNet, Fewshot-CIFAR100, Caltech-UCSD Birds-200–2011) have demonstrated that our proposed algorithm achieves consistent improvement over 6 baseline models (Matching Networks, Relation Networks, Prototypical Networks, MAML, Baseline++, Meta Baseline) and 11 state-of-the-art models (DC, TapNet, SNAIL, TADAM, MetaOptNet, CAN, CTM, DCEM, AFHN, LEO, AWGIM). Our code is available at: (https://github.com/rumorgin/MIAN).

1. Introduction

With the emergence of Convolutional Neural Network (CNN), the accuracy of image classification tasks has been greatly improved. However, the CNN always need a mass of labeled images to train the model. For many practical problems such as biology and medicine, it is very hard to get enough images. In contrast, the human visual system can quickly learn a new concept from a limited number of samples based on prior knowledge. Inspired by this human learning system, the few-shot learning is proposed to learn a new class with only a little labeled data.

A key idea in few-shot learning is embedding each sample into a lower-dimensional space, making similar samples closer to each other while dissimilar samples are highly separated. Then it adopts some non-parametric metric classify methods such as nearest-neighbor to get the predicted label of test data in this space. This kind of few-shot learning method is also

* Corresponding author at: No. 2006, Xiyuan Ave, West Hi-Tech Zone, 611731, Chengdu 611731, China.
E-mail addresses: qinzhili@outlook.com (Z. Qin), hanwangme@std.uestc.edu.cn (H. Wang), cobbinahben@std.uestc.edu.cn (C.B. Mawuli), weihan@std.uestc.edu.cn (W. Han), zhang_rui@std.uestc.edu.cn (R. Zhang), qinli.yang@uestc.edu.cn (Q. Yang), junmshao@uestc.edu.cn (J. Shao).

https://doi.org/10.1016/j.ins.2022.07.013
0020-0255/

called metric-based few-shot learning; there are two key components of this method: first is the metric that measures the similarity among samples, second is the embedding function that is used to generate an appropriate embedding space.

Existing metric-based methods usually embed the complete image to learn the feature map. However, in the real-world scenario, a picture has many other objects besides the object related to its label. In some extreme cases, uncorrelated objects occupy most of the image space, and the target only exists in a small part of the image. These irrelevant parts will lead the feature map to confound target information. Moreover, because the receptive field of CNN is restricted, some features learned by this CNN may only represent the irrelevant object. If the dataset has enough samples for one class, the model can avoid this problem by learning the feature map from other high-confidence images, but in the few-shot learning scenario, the dataset only has a few samples for each class, therefore, the irrelevant feature map will highly influence the quality of the embedded space.

To tackle this problem, the simplest strategy is adding an extra target setting process before the few-shot classification problem, such as using saliency maps [43] or hand-made part-based annotations [1] to anchor the label correlated area in the image. Whereas these methods need extra information, information is hard to obtain because of labor and material cost. Another kind of idea is based on the self-attention mechanism [13]. Actually, through the analysis of image attention maps, the high-value area of attention maps will still miss the actual target location; this may be because of the lack of enough samples to train a complex self-attention mechanism with a large number of parameters, the model is still over-fitting.

Enlightened by the recent researches about vision transformers, they decompose the original image into patches. Thus the feature extractor can extract finer-grained image features and also serve as a data augmentation tool to prevent over-fitting. Meanwhile, the image patches are naturally a multi-instance bag; many multi-instance learning methods use this approach to process image data. Instead of learning an attention map on the whole image, in this paper, we propose Multi-Instance Attention Network (MIAN) that redefines the original few-shot classification problem as a multi-instance learning problem. We then follow the bag-level multi-instance learning strategy to solve this problem. Learning an attention-map from image patches greatly decreases attention computational complexity, and the network can benefit from multi-instance learning to accurately localize target regions from a larger spatial dimension rather than a pixel dimension. Specifically, as shown in Fig. 1, we divide the image into patches as grid-style; each patch of one image can be regarded as an instance, the whole image as the bag of instances. For these patches in the image, we assume that there exist individual labels, but these labels remained unknown during training. Some patches may have the same label as the image, some maybe not, but at least one patch's label is the same as the image. Considering the goal of multi-instance learning as predicting the label of bags, we follow the bag-level strategy that combined all of the instance information to learn the bag representation. These representations should be more similar to the patches which have the same label as the image, and dissimilar with these patches having different labels. The multi-instance learning problem now converts back to few-shot. We adapt the prototypical network to generate a higher-level concept for each class based on these bag representations. The label of a test image is predicted by the nearest high-level concept. To the best of our knowledge, our approach is the first to apply multi-instance learning to the few-shot learning task.

The contributions of this work can be summarized as follows: i) We provide a new point of view to deal with few-shot image classification. Specifically, we redefine this problem as a multi-instance learning problem. ii) We introduce a simple but intuitive attention-based instance-level multi-instance feature extractor to learn the representative features of the image data and transform the multi-instance feature to a few-shot learning scenario. iii) Our self-attention module does not belong to any traditional computer vision attention category; we proposed a new patch-level attention mechanism, which greatly reduces the computational complexity. Fig. 2.

The rest of this paper is organized as follows. First, Section 2 briefly reviews related works in few-shot learning, multi-instance learning, and attention mechanism. Section 3 presents our proposed multi-instance attention network. Section 4 presents the implementation details, experimental results, and model analysis. Finally, in Section 5 we conclude this article.

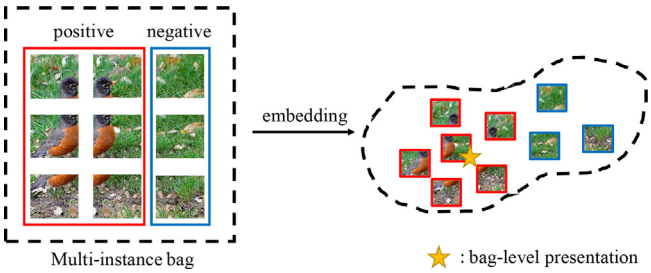

Fig. 1. Illustration of how to transform a few-shot learning problem into a multi-instance learning problem. The image comes from the *mini*imagenet dataset, its corresponding label is *house finch*, one species of bird. We divide this picture into 9 parts as grid-style, all patches make up a multi-instance bag in the black dotted box. Although the exact label of each patch is unknown, our multi-instance learning network will softly recognize them by learning a bag-level presentation that extracts more positive patch features meanwhile eliminate negative patch information.

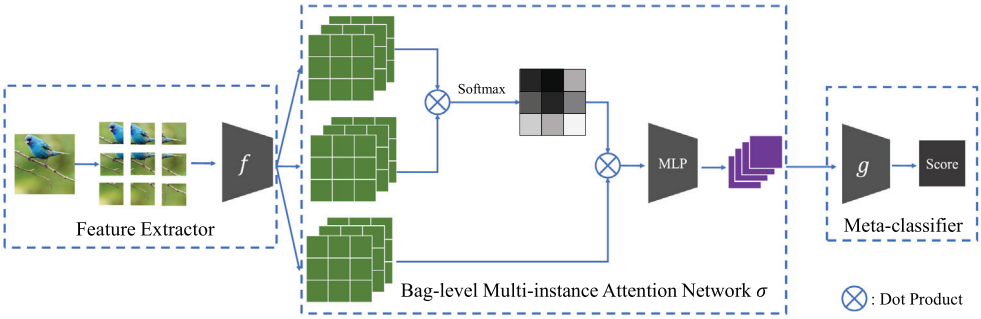

Fig. 2. The framework of the proposed method. Following the bag-level multi-instance learning rules, the method is divided into three parts. The first part includes image preprocessing and the feature embedding function f, which converts the image into a multi-instance bag. We divide it into 9 parts, and use a backbone CNN to generate the feature. The second part is the permutation-invariant aggregation function σ, which learns the attention map to emphasize the part that includes label corresponding object, multiplies the feature matrix with the attention map to get the bag-level feature, then restores the multi-instance learning problem to a few-shot learning problem by a linear layer. The third part is the score function g, which classifies this image by a meta-classifier.

2. Related Work

Few-shot learning with attention mechanism. Attention mechanism is motivated by the human visual system. When humans observe one object, they focus on certain key points with high resolution while perceiving the surrounding place in low-resolution [21,17,8,24]. Coincidentally, few-shot learning is also insight by the neurological phenomenon of the human, which is the human do not need a large number of samples to learn a particular concept, but only to reason about existing similar concepts [33,2,29]. Combining these two domains can mimic more complex human behavior, and applying them to few-shot learning can lead to higher model accuracy.

The Attention mechanism in few-shot learning can be divided into two categories: semantic-based attention and feature-based attention. The semantic-based attention generally needs extra semantic information, like the fine-grained image label, the semantic word embedding, and hierarchical semantic label. Based on this rich semantic information, the attention network can yield a more differentiated representation space. Yan et al. [40] designs a multi-task loss to embed the semantic similarity into different tasks. Cheraghian et al. [4] proposes an attention mechanism to align semantic vectors to the visual vectors, these semantic vectors are regarded as regularizer constraints to prevent catastrophic forgetting in few-shot incremental learning problems.

The feature-based attention is only based on original image features, the usual aim is to make the model focus on the part of the image that is most relevant to the label. Most computer vision methods extract image features by CNN, and the CNN-generated features usually have three dimensions: weight, height, and channel. The channel dimension is dependent on convolutional kernels, which are basic feature detectors in CNN. Thus, channel attention can help CNN learn the meaningful representations of the image input [36,11]. Matching Net [35] is the first method to introduce channel attention module into few-shot learning, they design a cosine distance style attention module to guide classifying. Dense Classification (DC) [23] separates the full feature tensor into pieces of channel-wised vectors, then parallelly input them in a parameter-shared attention module to learn a representative feature embedding.

The weight and height dimensions of feature vectors correspond with the original images' weight and height. Therefore, the spatial attention on weight and height dimensions controls where to pay attention [8,24]. Integrated spatial attention into few-shot learning, Hou et al. creates the Cross Attention Network (CAN) to generate cross attention maps for a pair of support feature and query feature, which highlighted similarities in the pair of features. Li et al. [20] proposes the Category Travesal Module (CTM) which can learn the inter-class commonality and inter-class uniqueness by the attention-based concentrator and projector.

Our proposed attention mechanism is closed to the feature-based attention. We split the original image into patches, extending a new dimension in image data, namely, the patch dimension. On the one hand, the number of patch dimensions is usually much smaller than the traditional three dimensions, thus greatly reducing the number of attention module parameters. On the other hand, the patch dimensional attention mechanism can benefit from multi-instance learning and achieve a good compromise between global and local features. Our model can also benefit from Visual Transformer (ViT) [8] because of the strong similarity of image preprocessing between ViT and our method. Specifically, both of them require image cropping, and in the attention mechanism part, we adopt a part of the transformer structure and the multi-head self-attention network to implement our network.

Multi-instance representation. Multi-instance learning is a form of weakly supervised learning. Each instance in multi-instance learning does not have a definite label, and only a structure called a bag composed of multiple instances with label information. If a bag has a positive label, it means that at least one instance in the bag is positive, and if a bag has a negative label, it indicates all of the instances in this bag are negative. The goal of multi-instance learning is predicting the label of the

test bag. Based on the representation level, the multi-instance learning can be divided into two categories, one is the instance-level method, the other is the bag-level.

The instance-level method aims to transfer the multi-instance learning problem to the traditional classification problem. Specifically, it tries to learn the label for each instance, then integrates the instance labels in the same bag to get the final label of the bag. APR [6] is the first proposed instance-level method to solve the multi-instance problem. The idea is to find the most representative location in the feature space, which should include positive instances as much as possible and exclude all of negative instances. Diverse density [25] inherits this idea; a formula based on probability is proposed to calculate the best representation concept. SP-B-MIL [39] adapts a self-paced loss into multi-instance learning to deal with the problem of only a small number of bags being labeled. However, the dataset of multi-instance has no instance labels at all, therefore it's hard to evaluate the accuracy of instance labels.

The bag-level method avoids this problem by directly predicting the label of the bag. Wei et al.[38] uses a vector of locally aggregated descriptors and fisher vector to encode bags into low-dimensional features, making the proposed methods suitable for large scale data. Küçükaşcı et al. [18] proposes a hash encoding module to obtain the bag feature. With the success of deep learning in various fields, Chi et al. [5] proposed two multilayer perceptron modules, which are instance feature block and bag classifier block to exploit the information in the negative bags. Ilse et al. [16] adds an attention module to the neural network as the permutation-invariant operators archived higher results. Using human-defined metric formulas to measure the similarity between bags usually has limited expressive power. BSN [37] adopts a neural network to automatically learn the similarity among bags, which improves the accuracy of the metric.

Collectively, multi-instance learning has good denoising ability, and can be used as a target positioning method without prior information in computer vision [31]. However, to the best of our knowledge, the multi-instance learning concept has not been introduced to the few-shot learning problem. Applying it to few-shot learning can achieve good target positioning results with minimal algorithm complexity cost, instead of training on a large number of manual labeling location information. Furthermore, one shortcoming of multi-instance learning is most of the methods only consider binary classification, so in our method, we extend the traditional multi-instance learning to multi-class classification scenarios.

At the end, we summarize the comparison between the proposed method and some existing related metric-based methods, see Table 1.

3. Method

3.1. Problem Definition

The episode training framework is the mainstream of the few-shot learning method, also known as the N-way K-shot framework. The original dataset needs to be divided into two new sets: base set $\mathscr{D}_{base}$ and novel set $\mathscr{D}_{novel}$. These two sets will be used in two stages called meta-train and meta-test, respectively. The class of $\mathscr{D}_{base}$ and $\mathscr{D}_{novel}$ are disjoint, namely $\mathscr{D}_{base} \cap \mathscr{D}_{novel} = \varnothing$. In the meta-train stage, we build a large number of episodes from $\mathscr{D}_{base}$. Each episode is a classical classification problem, consisting of training and testing two sub-stages. An episode $\mathscr{T}_i$randomly samples N classes of data. Each class has K samples, thus $N \times K$ samples are called the support set $\mathscr{S} = \{(x_i, y_i)\}_{i=0}^{N\times K}$, which is used for training the model. Meanwhile, we sampling from each of the N categories M samples, where $N \times M$ samples are called the query set $\mathscr{Q} = \{(x_i, y_i)\}_{i=0}^{M}$, used for testing. Here, x_i denotes the image sampled from the dataset, y_i denotes the corresponding label of x_i. At the meta-test stage, the way to build the episode is the same as the meta-train stage. The sole difference is that the label of the query set sample is unknown. The few-shot learning model needs to use the limited labeled sample in support set to predict the label of samples in the query set. A variables table of our method is shown at Table 2.

Table 1
The comparison between the proposed method and some existing related metric-based methods. Here FT denotes whether a fine-tuning strategy is used, Atten denotes what attention mechanism is employed(C: channel-wised attention; S: spatial-wised attention; P: patch-wised attention), Encoder means the type of feature extractor, Similarity is the similarity metric used for classification.

Method	Year	Author	FT	Atten	Encoder	Similarity
Matching Net[35]	2016	Vinyals et al.	No	C	4conv	Cosine
Relation Net[32]	2018	Sung et al.	No	No	4conv	Network learning
Prototypical Net[30]	2017	Snell et al.	No	No	4conv	Squared Euclidean
Baseline++[2]	2018	Chen et al.	Yes	No	4conv	Cosine
DC[23]	2019	Lifchitz et al.	Yes	C	ResNet12	Scaled cosine
TapNet[41]	2019	Yoon et al.	No	No	ResNet12	Network learning
TADAM[27]	2018	Oreshkin et al.	No	No	ResNet12	Squared Euclidean
Meta-Baseline[3]	2021	Chen et al.	Yes	No	ResNet12	Scaled cosine
CAN[15]	2019	Hou et al.	No	S, C	ResNet12	Cosine
CTM[20]	2019	Li et al.	No	S, C	ResNet18	Network learning
DCEM[9]	2019	Dvornik et al.	Yes	No	ResNet18	Cosine
AFHN[22]	2020	Li et al.	Yes	No	ResNet18	Cosine
MIAN (Ours)			Yes	P	ResNet12	Squared Euclidean

Table 2
Frequently used notions.

Variables	Description
$\mathscr{D}_{base}$	Base data set
$\mathscr{D}_{novel}$	Novel data set
$\mathscr{T}_i$	A learning task of few-shot learning framework
$\mathscr{S}$	Support set
$\mathscr{Q}$	Query set
x_i	An image sample
y_i	The label corresponding to image
P	The original image
X	The feature matrix of one multi-instance bag
Q, K, V	The query, key, value matrix of self-attetnion module
A	The self-attention matrix of multi-instance bag
H	The aggregation matrix
$\widehat{X}$	The aggregated feature embedding of multi-instance bag
c_j	The prototype of class j

Table 3
Abbreviations of methods.

Abbreviations	Description
AFHN	**A**dversarial **F**eature **H**allucination **N**etwork
AWGIM	**A**ttentive **W**eights **G**eneration via **I**nformation **M**aximization
CAN	**C**ross **A**ttention **N**etwork
CTM	**C**ategory **T**ravesal **M**odule
DC	**D**ense **C**lassification
DCEM	**D**iversity with **C**ooperation **E**nsemble **M**ethods
LEO	**L**atent **E**mbedding **O**ptimization
MAML	**M**odel-**A**gnostic **M**eta-**L**earning
MIAN	**M**ulti-**I**nstance **A**ttention **N**etwork
SNAIL	A **S**imple **N**eural **A**ttent**I**ve Meta-**L**earner
TADAM	**TA**sk **D**ependent **A**daptive **M**etric
TapNet	**T**ask-**A**daptive **P**rojection **Net**work

3.2. Multi-class Multi-instance Attention Network

Classical multi-instance learning only focuses on binary classification problems. This is enough for the molecule activity scenario, but for image classification problems, there are usually a large number of categories that need to be distinguished by the model. Compared with the standard multi-instance learning assumption, which is that all negative bags only contain negative instances. Meanwhile, positive bags contain at least one positive instance. Here we only assume that the multi-class multi-instance learning has at least one instance in a bag that has the same label as the category of this bag. With this assumption, we can extend the binary multi-instance learning to the multi-class classification problem. But if we consider this assumption in an instance-based view, it will lead to a new problem. Thus, from the perspective of binary multi-instance learning, for a given bag including K instances, one instance in this bag must belong to either the positive class or the negative class. However, if there are more than two classes in one bag, signifying the label no longer represents only the logical information of *True* or *False*, but represents the semantic information of a main object in the instance. To this end, if one specific instance only includes the background, there will be no label corresponding to this instance. Fortunately, the bag-level method saves us from considering the impact of background instances. This method focuses on learning a low-dimensional embedding from these instances belonging to the same bag. This embedding should be representative of the corresponding class' bag while reducing the effect of irrelevant instances.

To emphasize the instance information of the corresponding category in the bag while suppressing irrelevant information, based on the classical multi-head attention network, we propose the multi-class multi-instance attention network. First, as a multi-instance network, besides playing the original role of the attention mechanism, the network also follows the multi-instance learning rules. The most important rule of the bag-level method is that the bag probability must be permutation-invariant. To ensure this rule, the bag probability should follow Theorem 1 [42].

Theorem 1. *A scoring function for a set of instances $X, S(X) \in \mathbb{R}$is permutation-invariant to the elements in X, iff it can be decomposed in the following form:*

$$S(X) = g(\sigma\{f(x_1), \ldots, f(x_n)\})$$

where f and g are suitable transformations, σ is pooling operator of neural network.

This rule requires that the bag-level method must have three characters: i) an embedding function f encoding instances into low-dimensional features; ii) a permutation invariance aggregation function σ gathering features; iii) an embedding function g predicting the final classification score based on the aggregated features of the previous step. In our method, f is the CNN feature extractor, σ is our proposed attention network, and the score function g is adapted from the prototypical network.

Feature embedding function f**.** Before extracting the features, we first need to transform the few-shot learning problem into a multi-instance learning problem. The basic idea is to split the full image into patches, then we can treat the full image as a multi-instance bag, and each part is the instance of this bag. There are many ways to split the full image, such as random cropping and grid cropping. According to other visual transformer methods, here we divide the full image into 9 parts as grid-style. Considering that the original image size of some datasets is small, we enlarge the size of each block so that there is a slight overlap between them to get enough information for each sample. Formally, Let $P \in \mathbb{R}^{d_H \times d_W \times d_C}$ denotes one full image, here d_H, d_W and d_C are height, weight and channel of the image, respectively. If we cut the image into 3×3 patches using a grid-based approach, each patch P_i should has size of $d_H/3 \times d_W/3 \times d_C$. And we enlarge patches on height and weight to let patches have more information by adding a constant on them. The dimensions change to $(d_H/3 + \delta) \times (d_W/3 + \delta) \times d_C$.

After converting the original image into a multi-instance learning bag, these patches only include the local information of the full image. But the global information of the picture is also very important, such as the shape information of the object, so we add an extra resized full image in bags as another instance. Following the bag-level method of three characters, we choose one backbone CNN as the feature embedding function f. Each image patch is then passed through f to generate the features.

Permutation-invariant aggregation function σ**.** There are two traditional σ in bag-level methods: maximum operator and mean operator. The maximum operator selects the highest confidence instance feature in the current bag as the bag feature. During model training, the operator can indeed gradually select the correct positive instance as the bag representation, but the information of the remaining samples will be completely lost. On the contrary, the mean operator combines all instances information into bag representation without considering instance confidence. That is to say, instances of negative classes are also integrated into the bag feature, which will greatly impair the precision of the feature. The attention network combines the advantages of these two primitive operators, assigns an importance coefficient to each instance in the bag, then integrates the information of all instances by a learnable method.

The input of the attention network is the output of f; this feature matrix contains all feature vectors of instances in this bag. Let $X \in \mathbb{R}^{d_D \times d_K}$ be the feature matrix, d_D is the feature dimension of one instance, d_K is the number of instances in the bag. Without multi-head, a single head self-attention network should have three learnable parameter matrices: $W^Q, W^K, W^V \in \mathbb{R}^{d_K \times d_K}$. Here the matrices work on instances, so the dimension is d_K instead of the original single-head attention dimension of d_D. This operation also can significantly reduce the number of model parameters. Then we multiply the input with the three learnable parameters to get three different new representations $Q, K, V \in \mathbb{R}^{d_D \times d_K}$:

$$Q = XW^Q, K = XW^K, V = XW^V \tag{1}$$

Now we define the self attention $A \in \mathbb{R}^{d_K \times d_K}$ in scaled dot product form:

$$A = softmax\left(\frac{Q^T K}{\sqrt{d_K}}\right) \tag{2}$$

In the next step, we get the weighted feature matrix of the instances by multiplying V and A. Until now the number of instances is still d_K, the aggregation function should reduce this number to 1 to get the final bag representation. So we add one extra learnable parameter matrix $H \in \mathbb{R}^{d_K \times 1}$ to get the weighted summation of the feature matrix of the sample $\widehat{X} \in \mathbb{R}^{d_D \times 1}$. This process can be represented as:

$$\widehat{X} = VAH \tag{3}$$

Splitting the single-head attention into multi-head attention can enhance the local representation ability of the model. Let the number of heads be h, so we need to divide X into h parts along with feature dimension: $X = \{x_1, x_2, \ldots, x_h\}, x_i \in \mathbb{R}^{d_K \times d_D/h}$. Correspondingly, for each part of the original input, we should create three learnable parameter matrices: $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_K \times d_K}$, these matrices make the model pay attention to different aspects of information. Before the step of aggregating the weighted input matrix, we need to concatenate each head-part feature into one full feature matrix. The remaining processes are the same as the single-head attention network.

Overall, our attention network can be considered as a weighted average operator, meeting the requirements of the invariant model in [42].

Score function g**.** With the bag-level feature obtained, our problem is redefined as a classical few-shot learning problem, or in multi-instance view. The only remaining operation we need is a score function g to predicting the label. We adopt a simple meta-classifier prototypical network to be our score function:

$$c_j = \frac{1}{|S_j|}\sum_{i\in S_j}\sigma(f(P_i)) \tag{4}$$

Here S_j denotes the set of image corresponding to class j in support set, $|S_j|$ denotes the number of image in S_j. The classification strategy of our model follows the nearest neighbor rule: let q denotes one of the query images, $\mathbb{P}(y=j|q)$ is the probability of q belongs to class j. The probability is measured by a softmax function:

$$\mathbb{P}(y=j|q) = \frac{\exp\left(-d\left(\sigma(f(q)), c_j\right)\right)}{\sum_{j\prime}\exp\left(-d\left(\sigma(f(q)), c_{j\prime}\right)\right)} \tag{5}$$

Here $j\prime$ denotes each class in the support set, and d is a metric function; we simply choose the Euclidean distance. Finally, for the loss function of the network, we minimize the negative log-probability function:

$$\mathscr{L} = -\log\mathbb{P}(y=j|q) \tag{6}$$

Similar as defined previously, $\mathbb{P}(y=j|q)$ is the probability of query images q belonging to the ground-truth class of query image j. If query images q has a high probability of belonging to class j, the value of $\mathbb{P}(y=j|q)$ will near to 1, and value of loss will be closed to 0, which means the model make an accurate prediction, vice versa.

3.3. Computational Complexity

We split our model into CNN module and multi-head attention module to analyze the computational complexity. The CNN module is a standard ResNet12 network, so based on [14], the time computational complexity is $\mathcal{O}\left(\sum_{l=1}^{D} M_l^2 K_l^2 C_{l-1} C_l\right)$, where D is the depth of ResNet12; l is the l-th layer of CNN;M_l, K_l and C_l is the length of feature map. The length of the convolutional kernel and the output channel number in l-th CNN layer, respectively. Our multi-head attention module is also similar to the standard multi-head attention. The time overhead is mainly on the linear transform Eq. 1 and the self-attentive mechanism Eq. 2. The time computational complexity is $\mathcal{O}\left(d_K^2 d_D + d_K d_D^2\right)$. Here the definition of d_K and d_D are the number of instances in one bag and the number of dimensions of one instance, respectively, as already discussed. In vanilla multi-head attention module, d_K is the number of instances of one batch, normally set to 64, 128, and so on. In our method, d_K is set to 10, which greatly reduces computational complexity.

4. Experiments

We conduct experiments on four popular datasets and compare our method with the state-of-the-art methods to evaluate our approach.

4.1. Implementation Details

Our experimental platform is a computer with Intel Xeon CPU E5-2678 v3 CPU, Nvidia GeForce RTX 3090 GPU and Ubuntu 18.04 system. At the data preprocessing, we divide the full image into 9 patches, and set the size of patches as 84×84. For each patch, we adapt random horizontal flip and color jitter as the data augmentation methods. For fair comparing, our backbone CNN is set as ResNet12 architecture, which is widely used in much few-shot learning methods[41,19]. This CNN includes four residual blocks, each block has three 3×3 convolutional layers, and owns one max pooling layer at the end. The output feature map shape is $640 \times 5 \times 5$. The attention module is designed similarly with the multi-head attention in transformer method[34]. For each single-head attention sub-module, W_i^Q, W_i^K, W_i^V are three same size linear layers. For the step of converting instance features into bag representation, we also use one linear layer for the implementation. Another technique in our method is pre-training. Following the idea of [2], we use all the meta-training samples to train the backbone and validate using the 5-way 5-shot few-shot learning framework. At the meta-train stage, we set the max epoch to 120, use SGD as our optimizer and initialize the learning rate to 0.005. We applied a learning rate scheduling technique by multiplying the learning rate by 0.5 after every 40 episodes. Moreover, we validate our training model after every 50 batches. The validation step has 500 episodes, test step has more episodes to obtain a stable result, we set it to 5000 episodes. There are 3 important hyper-parameters in our method: the size of patches s, the number of patches in one instance K, the number of head in attention mechanism h. During the training process, we choose $s \in \{1, 1.5, 2\}, K \in \{1, 4, 9\}, h \in \{3, 6, 9\}$, all these parameters analysis are presented at Section 4.4.

4.2. Datasets

Our experiments are conducted on four popular benchmark datasets as follows.

*mini***ImageNet.** This is a mini-version of the famous ImageNet dataset, with sampling from ILSVRC-12. It contains 100 classes with 600 images for every class. The resolution of each image is 84×84. As usual, we split the 100 classes into 64, 16, and 20 classes used for training, validation, and testing, respectively.

*tiered***ImageNet.** This is also a subset of ILSVRC-12, its classes have a hierarchical structure, namely, the dataset is divided into 34 top categories(such as vehicles, musical instruments, etc.). Each top category contains 10 to 30 more detailed sub-categories(such as musical instruments including guitars, pianos, etc.). The top categories are divided into 20 training classes, 6 validation classes, 8 testing classes.

Fewshot-CIFAR100. This dataset has the organization of the same sample, it has 100 classes with each class having 600 images, but the resolution of images is only 32×32, this resolution makes the classification task harder than other datasets. We partition the classes as 60 for the training set, 20 for the validation set, and 20 for the testing set.

Caltech-UCSD Birds-200–2011. CUB is a fine-grained classification dataset, each image not only includes the label, but also has a bounding box, part location, and binary attribute information. The dataset consists of 11788 images in 200 categories. We split this dataset into 100, 50, and 50 for training, validation, and testing, respectively.

4.3. Comparison with Other Methods

For fair comparisons, we compare the results of other methods which use the same ResNet12 backbone as used in this work. If there are no reported results on ResNet12, we choose a deeper CNN backbone, such as ResNet18 or ResNet24. There are two kinds of popular experimental sets in few-shot learning: 5-way 1-shot and 5-way 5-shot classification. We report our classification accuracies with 95% confidence intervals on each set. For the comparison methods of *mini*imagenet dataset, we choose five methods as the baselines, namely matching networks, relation networks, prototypical networks, baseline++, and meta-baseline, all of these methods have simple structure but great performance. And the other methods are state-of-the-art, mostly they are metric-based methods. The results are shown in Table 4. For datasets that are not evaluated with some of the comparison methods, we only report the existing experimental results. The results are shown in Table 5–7. The abbreviations of methods are shown in Table 3.

As we can see, our approach has achieved better performance than existing methods in both experiments settings on four datasets. Especially on the 5-way 5-shot set, our method improves by an average of 1.6% compared to the second-best algorithm on all datasets. On the other hand, we can see our method always have better performance on 5-way 5-shot set than 5-way 1-shot set. Most of the results on 5-way 1-shot sets only performed slightly higher than the baselines. This may be because our multi-instance attention module has four parameters matrix to learn. Also the situation of providing one image for each class makes our module to over-fitting sometimes. Moreover, after obtaining the feature of the image bag, the remaining process of our method is basically the same as that of the prototypical network. Compared with the prototypical network on four datasets, we have improved the accuracy rate by 2.9% at the 5-way 1-shot set and 3.5% at the 5-way 5-shot set.

4.4. Ablation study and discussion

We conduct the ablation study on all datasets to confirm the contribution of different components of our method. For convenience, we only conduct on the 5-way 5-shot set for this study.

Table 4
Results on *mini*ImageNet dataset.

Model	Backbone	Type	5-way 1-shot	5-way 5-shot
Matching Networks[35]	ResNet12	Metric	63.08 ± 0.80	75.99 ± 0.60
Relation Networks[32]	ResNet12	Metric	52.19 ± 0.83	70.20 ± 0.66
Prototypical Networks[30]	ResNet12	Metric	60.37 ± 0.83	78.02 ± 0.57
MAML[10]	ResNet12	Optimization	54.69 ± 0.89	66.62 ± 0.83
Baseline++[2]	ResNet12	Metric	53.97 ± 0.79	75.90 ± 0.61
DC[23]	ResNet12	Metric	62.53 ± 0.19	79.77 ± 0.19
TapNet[41]	ResNet12	Metric	61.65 ± 0.15	76.36 ± 0.10
SNAIL[26]	ResNet12	Model	55.71 ± 0.99	68.88 ± 0.92
TADAM[27]	ResNet12	Metric	58.50 ± 0.30	76.70 ± 0.30
Meta Baseline[3]	ResNet12	Metric	63.17 ± 0.23	79.26 ± 0.17
MetaOptNet[19]	ResNet12	Model	62.64 ± 0.82	78.63 ± 0.46
CAN[15]	ResNet12	Metric	63.85 ± 0.48	79.44 ± 0.34
CTM[20]	ResNet18	Metric	64.12 ± 0.82	80.51 ± 0.13
DCEM[9]	ResNet18	Metric	58.71 ± 0.62	77.28 ± 0.46
AFHN[22]	ResNet18	Metric	62.38 ± 0.72	78.16 ± 0.56
LEO[28]	WRN-28	Optimization	61.76 ± 0.08	77.59 ± 0.12
AWGIM[12]	WRN-28	Optimization	63.12 ± 0.08	78.40 ± 0.11
MIAN (Ours)	ResNet12	Metric	**64.27 ± 0.35**	**81.24 ± 0.26**

Table 5
Results on *tiered*ImageNet dataset.

Model	Backbone	Type	1-shot	5-shot
Matching Networks[35]	ResNet12	Metric	68.50 ± 0.92	80.60 ± 0.71
Relation Networks[32]	ResNet12	Metric	64.42 ± 0.36	81.74 ± 0.61
Prototypical Networks[30]	ResNet12	Metric	65.65 ± 0.92	83.40 ± 0.65
TapNet[41]	ResNet12	Metric	63.06 ± 0.15	80.26 ± 0.12
Meta Baseline[3]	ResNet12	Metric	68.62 ± 0.27	83.74 ± 0.18
MetaOptNet[19]	ResNet12	Model	65.99 ± 0.72	81.56 ± 0.53
CAN[15]	ResNet12	Metric	69.89 ± 0.51	84.23 ± 0.37
CTM[20]	ResNet18	Metric	68.41 ± 0.39	84.28 ± 1.73
LEO[28]	WRN-28	Optimization	66.33 ± 0.05	81.44 ± 0.09
AWGIM[12]	WRN-28	Optimization	67.69 ± 0.11	82.82 ± 0.13
MIAN (Ours)	ResNet12	Metric	**69.89 ± 0.36**	**86.05 ± 0.26**

Table 6
Results on Caltech-UCSD Birds-200–2011 dataset.

Model	Backbone	Type	1-shot	5-shot
Matching Networks[35]	ResNet12	Metric	71.29 ± 0.87	83.47 ± 0.58
Relation Networks[32]	ResNet12	Metric	70.47 ± 0.99	83.70 ± 0.55
Prototypical Networks[30]	ResNet12	Metric	71.22 ± 0.92	85.01 ± 0.52
Baseline++[2]	ResNet12	Metric	69.55 ± 0.89	85.17 ± 0.50
MAML[10]	ResNet12	Optimization	70.32 ± 0.99	80.93 ± 0.71
AFHN[22]	ResNet18	Metric	70.53 ± 1.01	83.95 ± 0.63
MIAN (Ours)	ResNet12	Metric	**71.86 ± 0.35**	**85.84 ± 0.23**

Table 7
Results on Fewshot-CIFAR100 dataset.

Model	Backbone	Type	1-shot	5-shot
Matching Networks[35]	ResNet12	Metric	43.88 ± 0.75	57.05 ± 0.71
Relation Networks[32]	ResNet12	Metric	42.41 ± 0.21	57.23 ± 0.62
Prototypical Networks[30]	ResNet12	Metric	41.54 ± 0.76	57.08 ± 0.76
DC[23]	ResNet12	Metric	42.04 ± 0.17	57.63 ± 0.23
TADAM[27]	ResNet12	Metric	40.10 ± 0.40	56.10 ± 0.40
MIAN (Ours)	ResNet12	Metric	**44.54 ± 0.33**	**58.09±0.32**

Components Analysis. There are two key components of our method: the image cropping process and multi-instance multi-head attention module. We create three-stages experiments by adapting them into our baselines. The first stage is the baseline; our method is based on prototypical networks, therefore we set it as the baseline. The second stage is only to adapt the image cropping process into the baseline. This process will transform the classical classification problem into a multi-instance learning problem, namely, each cropped patch has no label. We simply average all patches in one bag as the bag feature. From the results of the four datasets, we see the cropping process achieves 0.2% average accuracy compared with the baseline. The process can be regarded as one data augmentation strategy, that can slightly increase the result. Because our attention module only fits for the multi-instance scenario, it is impossible to directly adapt it into the baseline without transforming the original problem into a multi-instance learning problem. For this reason, the last stage is the full model of our method. Compared with the stage two process, we can see the attention module makes a great improvement on the accuracy of all datasets. The results are shown in Table 8.

Size of patches. We consider three kinds of patch sizes: the first one comes from the idea of patches-based self-supervised learning method [7]. They first cut the full image into parts as the grid-style, then shrinks the length and width of patches to increase the learning difficulty of the model. The second method is the compromise of our method

Table 8
Results for components analyse, all results are obtained on the 5-way 5-shot set.

Model	*mini*imagenet	*tiered*imagenet	CUB	FC100
Baseline	78.02 ± 0.57	83.40 ± 0.65	85.01 ± 0.52	57.08 ± 0.76
Data augmentation	78.04 ± 0.27	84.61 ± 0.27	85.78 ± 0.25	57.19 ± 0.33
Full model	81.24 ± 0.26	86.05 ± 0.26	85.84 ± 0.23	58.09 ± 0.32

and the self-supervised learning method; just divides the image into grid-style parts exactly. The third method is our proposed method; add half the length and width to exactly blocks to obtain more information for little size original images. We conduct this analysis on *mini*ImageNet datasets of 5-way 1-shot set, and design a patch ratio hyper-parameter of our method to control the size of patches, we set its value as 1, 1.5, 2 to implement these three kinds of patch sizes. The results are shown in Table 9. As we can see, the larger the patch size, the higher the accuracy of the model. But compared to the prototypical network which extracts features from the whole image, patch features can capture more distinctive features and get better results.

Number of patches. We conduct this analysis on *mini*ImageNet datasets of 5-way 1-shot set, all parameters of this experiment are the same as we declared in Section 4.1 except the number of patches of each image K. Due to the limitation of GPU memory, we set $K \in \{1, 4, 9\}$, as results shown in Table 9. The accuracy of the model continues to improve as the number of patches increases, but since each patch is scaled to a size of 84×84, which also leads to a linear increase in memory overhead. Under the constraints of this experimental environment, we choose the number of patches to be 9.

Number of heads in multi-head attention. Changing the number of heads in multi-head attention will not influence the computational complexity, but will resize the feature dimension of self-attention network. Different head attention focuses on different parts of image, and learns different patterns. As the number of heads increases, the attention module obtains more semantic information from features. The experiment set is the same as **Number of patches**. The results are shown in Table 9. It can be seen that the model results are relatively stable for the variation of the attention head. It may be that the features input to the attention module are themselves already part of the original image with less information, and further segmentation of the features cannot bring too much gain.

Attention map visualization. To verify the effectiveness of our attention mechanism, we visualized the weight of the attention mechanism.

Fig. 3. All results are obtained from the query set images of the *mini*ImageNet dataset. It can be seen that the image patches containing the target object always have higher weights.

Table 9
Results of model hyper-parameters, all results are obtained on the 5-way 1-shot set of *mini*ImageNet.

Size of patches	**1**	**1.5**	**2**
*mini*ImageNet(5-way 1-shot)	58.43 ± 0.26	61.52 ± 0.25	64.27 ± 0.35
Number of patches	**1**	**4**	**9**
*mini*ImageNet(5-way 1-shot)	60.14 ± 0.34	62.83 ± 0.34	64.27 ± 0.35
Number of heads	**2**	**4**	**8**
*mini*ImageNet(5-way 1-shot)	63.08 ± 0.34	63.59 ± 0.34	64.27 ± 0.35

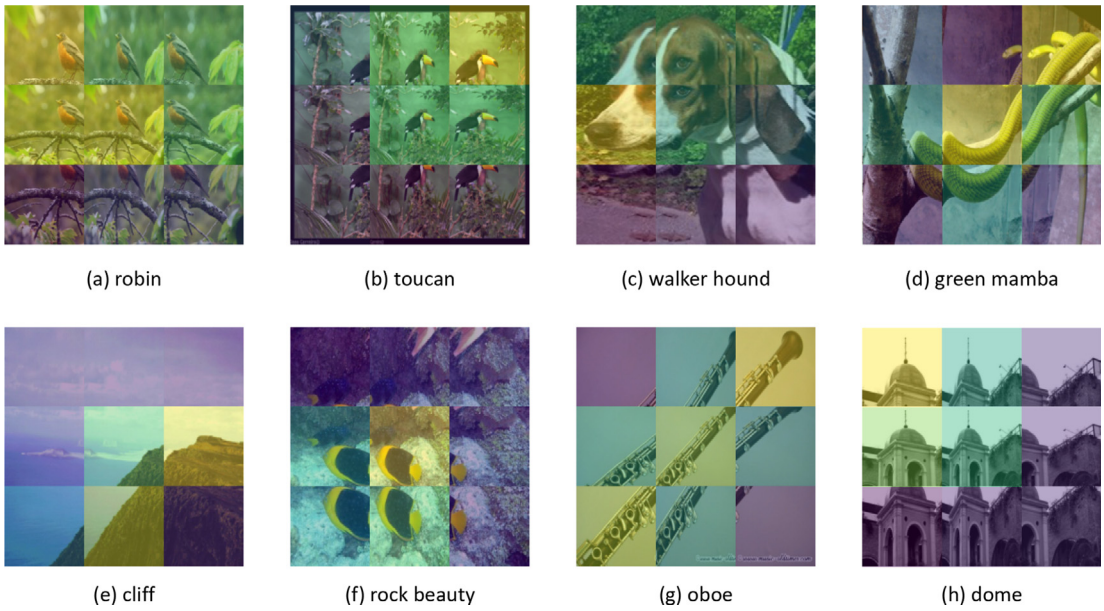

Fig. 3. Attention map of query image in *mini*imagenet dataset. Here the color bar of heat map is viridis style, the figure notes are the semantic label of images.

5. Conclusions and Future Work

We observe that existing few-shot learning methods based on the attention mechanism cannot match the feature map to the target region accurately. In this paper, we propose a novel few-shot learning method that transforms the original few-shot learning problem into a multi-instance learning problem. By transforming each image into a multi-instance bag, we design a multi-instance based multi-head attention module to obtain large-scale attention map to prevent over-fitting, and significantly reduces the computational complexity compared with recent attention networks. The experiments demonstrate that our method is competitive and outperforms most of the state-of-the-art methods. However, there is still much space for improvement in our approach. Our proposed feature extractor simply segments the image and then extracts the features of each patch, whereas the relative position between different patch is also a very important target information. Therefore, a feature extractor that incorporates self-supervised learning can be considered: in addition to the current few-shot classification task of this method, a jigsaw puzzle task can be constructed for the segmented patches of a picture, where each patch is given a relative position label and predicted. Thus, the relative position information is also incorporated into the features. Another improved point is maintaining a multi-instance metric space for bags. Designing a metric loss that keeps same classes bags close to each other and different classes bags far away can also theoretically improve the quality of bag features significantly. The above is the improvement for few-shot learning, while our proposed method can also solve the multi-instance learning problem with little change. Our current model input is based on the meta-learning framework, for example, input 5-way 5-shot samples to create a classification task. To tackle the multi-instance learning problem, we need to change the logic of input samples to the random batch approach.

CRediT authorship contribution statement

Zhili Qin: Conceptualization, Methodology, Software. **Han Wang:** Data curation, Formal analysis. **Cobbinah Bernard Mawuli:** Writing - review & editing. **Wei Han:** Software, Validation. **Rui Zhang:** Software, Validation. **Qinli Yang:** Writing - review & editing. **Junming Shao:** Supervision.

Declaration of Competing Interest

It is to certify that all authors have seen and approved the final version of the manuscript. They warrant that this manuscript is our original work, hasn't received prior publication and isn't under consideration for publication elsewhere.

This work is supported by the Fundamental Research Funds for the Central Universities (ZYGX2019Z014), National Natural Science Foundation of China (61976044, 52079026), Fok Ying-Tong Education Foundation (161062), and Sichuan Science and Technology Program (2020YFH0037).

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (ZYGX2019Z014), National Natural Science Foundation of China (61976044, 52079026), Fok Ying-Tong Education Foundation (161062), and Sichuan Science and Technology Program (2022YFG0260, 2020YFH0037).

References

[1] K. Cao, M. Brbic, J. Leskovec, Concept learners for few-shot learning, in: International Conference on Learning Representations, 2020.
[2] W.Y. Chen, Y.C. Liu, Z. Kira, Y.C.F. Wang, J.B. Huang, A closer look at few-shot classification, in: International Conference on Learning Representations, 2018.
[3] Y. Chen, Z. Liu, H. Xu, T. Darrell, X. Wang, Meta-baseline: Exploring simple meta-learning for few-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9062–9071.
[4] A. Cheraghian, S. Rahman, P. Fang, S.K. Roy, L. Petersson, M. Harandi, Semantic-aware knowledge distillation for few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2534–2543.
[5] Z. Chi, Z. Wang, W. Du, Explicit metric-based multiconcept multi-instance learning with triplet and superbag, in: IEEE Transactions on Neural Networks and Learning Systems, 2021.
[6] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1997) 31–71.
[7] C. Doersch, A. Gupta, A.A. Efros, Unsupervised visual representation learning by context prediction, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1422–1430.
[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2020.
[9] N. Dvornik, C. Schmid, J. Mairal, Diversity with cooperation: Ensemble methods for few-shot classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3723–3731.
[10] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.
[11] Z. Gao, J. Xie, Q. Wang, P. Li, Global second-order pooling convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3024–3033.
[12] Y. Guo, N.M. Cheung, Attentive weights generation for few shot learning via information maximization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13499–13508.

[13] F. Hao, F. He, J. Cheng, L. Wang, J. Cao, D. Tao, Collect and select: Semantic alignment metric learning for few-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8460–8469.
[14] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5353–5360.
[15] R. Hou, H. Chang, B. Ma, S. Shan, X. Chen, Cross attention network for few-shot classification, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 4003–4014.
[16] M. Ilse, J. Tomczak, M. Welling, Attention-based deep multiple instance learning, in: International conference on machine learning, PMLR, 2018, pp. 2127–2136.
[17] Y. Ji, H. Zhang, Z. Jie, L. Ma, Q.J. Wu, Casnet: A cross-attention siamese network for video salient object detection, IEEE Transactions on Neural Networks and Learning Systems 32 (2020) 2676–2690.
[18] E.Ş. Küçükaşci, M.G. Baydoğan, Bag encoding strategies in multiple instance learning problems, Information Sciences 467 (2018) 559–578.
[19] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10657–10665.
[20] H. Li, D. Eigen, S. Dodge, M. Zeiler, X. Wang, Finding task-relevant features for few-shot learning by category traversal, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1–10.
[21] J. Li, J. Wang, Q. Tian, W. Gao, S. Zhang, Global-local temporal representations for video person re-identification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3958–3967.
[22] K. Li, Y. Zhang, K. Li, Y. Fu, Adversarial feature hallucination networks for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13470–13479.
[23] Y. Lifchitz, Y. Avrithis, S. Picard, A. Bursuc, Dense classification and implanting for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9258–9267.
[24] Z. Lin, Z. He, S. Xie, X. Wang, J. Tan, J. Lu, B. Tan, Aanet: Adaptive attention network for covid-19 detection from chest x-ray images, IEEE Transactions on Neural Networks and Learning Systems 32 (2021) 4781–4792.
[25] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, Advances in Neural Information Processing Systems (1998) 570–576.
[26] N. Mishra, M. Rohaninejad, X. Chen, P. Abbeel, A simple neural attentive meta-learner, in: International Conference on Learning Representations, 2018.
[27] B.N. Oreshkin, P. Rodriguez, A. Lacoste, Tadam: task dependent adaptive metric for improved few-shot learning, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 719–729.
[28] A.A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, in: International Conference on Learning Representations, 2018.
[29] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: International conference on machine learning, PMLR, 2016, pp. 1842–1850.
[30] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4080–4090.
[31] P. Sudharshan, C. Petitjean, F. Spanhol, L.E. Oliveira, L. Heutte, P. Honeine, Multiple instance learning for histopathological breast cancer image classification, Expert Systems with Applications 117 (2019) 103–111.
[32] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1199–1208.
[33] P. Tian, W. Li, Y. Gao, Consistent meta-regularization for better meta-knowledge in few-shot learning, in: IEEE Transactions on Neural Networks and Learning Systems, 2021.
[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems (2017) 5998–6008.
[35] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al, Matching networks for one shot learning, Advances in Neural Information Processing Systems 29 (2016) 3630–3638.
[36] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, Eca-net: Efficient channel attention for deep convolutional neural networks, 2020. arXiv:1910.03151.
[37] X. Wang, Y. Yan, P. Tang, W. Liu, X. Guo, Bag similarity network for deep multi-instance learning, Information Sciences 504 (2019) 578–588.
[38] X.S. Wei, J. Wu, Z.H. Zhou, Scalable algorithms for multi-instance learning, IEEE transactions on neural networks and learning systems 28 (2016) 975–987.
[39] Y. Xiao, X. Yang, B. Liu, A new self-paced method for multiple instance boosting learning, Information Sciences 515 (2020) 80–90.
[40] S. Yan, S. Zhang, X. He, et al, A dual attention network with semantic embedding for few-shot learning, in: AAAI, 2019, pp. 9079–9086.
[41] S.W. Yoon, J. Seo, J. Moon, Tapnet: Neural network augmented with task-adaptive projection for few-shot learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 7115–7123.
[42] M. Zaheer, S. Kottur, S. Ravanbhakhsh, B. Póczos, R. Salakhutdinov, A.J. Smola, Deep sets, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 3394–3404.
[43] H. Zhang, J. Zhang, P. Koniusz, Few-shot learning via saliency-guided hallucination of samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2770–2779.