*Computers, Materials & Continua*
DOI: 10.32604/cmc.2023.027267
***Article***



# Few-Shot Object Detection Based on the Transformer and High-Resolution Network

**Dengyong Zhang[1,2], Huaijian Pu[1,2], Feng Li[1,2,*], Xiangling Ding[3] and Victor S. Sheng[4]**

[1]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, 410114, China
[2]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China
[3]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411004, China
[4]Department of Computer Science, Texas Tech University, Lubbock, 79409, TX, USA
* Corresponding Author: Feng Li. Email: lif@csust.edu.cn
Received: 14 January 2022; Accepted: 17 May 2022

**Abstract:** Now object detection based on deep learning tries different strategies. It uses fewer data training networks to achieve the effect of large dataset training. However, the existing methods usually do not achieve the balance between network parameters and training data. It makes the information provided by a small amount of picture data insufficient to optimize model parameters, resulting in unsatisfactory detection results. To improve the accuracy of few shot object detection, this paper proposes a network based on the transformer and high-resolution feature extraction (THR). High-resolution feature extraction maintains the resolution representation of the image. Channels and spatial attention are used to make the network focus on features that are more useful to the object. In addition, the recently popular transformer is used to fuse the features of the existing object. This compensates for the previous network failure by making full use of existing object features. Experiments on the Pascal VOC and MS-COCO datasets prove that the THR network has achieved better results than previous mainstream few shot object detection.

**Keywords:** Object detection; few shot object detection; transformer; high-resolution

## 1 Introduction

Since regions with convolutional neural network features (RCNN) [1] and You only look once (YOLO) [2], the two major networks of object detection, have been proposed, object detection has spawned many excellent papers. But these networks require an enormous amount of picture data for needing results. Labeling samples will generate higher production costs and some samples themselves are difficult to obtain, which caused the shortcomings of artificial intelligence to a certain extent

and hindered the research and development of artificial intelligence. Therefore, to tackle with the shortcomings, the use a small amount of training data to detect objects is meaningful.

Humans can recognize this category when only seeing a small number of samples, and we believe that neural networks should also have this ability. Given enough base class objects, a new class with only a few samples is used to detect the objects. As shown in Fig. 1, the training is divided into two stages: base training and fine-tuning training. In the base training, sufficient data training samples are used to construct a feature space that can represent the object feature. In the fine-tuning training, fine-tuning the network model contributes to the representation of the new class in the feature space. Few-shot object detection via feature reweighting (FSODFR) [3] is the first research to study few-shot object detection, which adjusts the meta-feature detection by the object vector. After that, two phases of meta-learning to detect rare objects (MetaDet) [4] and towards general solver for instance-level low-shot learning [5] have been proposed.

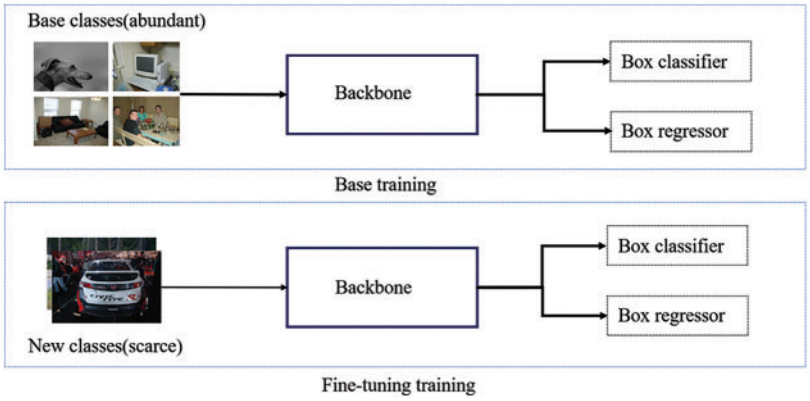


**Figure 1:** Few shot object detection methods based on fine-tuning. The figure above shows the base training, and the figure below shows the fine-tuning of the network trained

FSODFR [3] introduced a meta-feature learner and a lightweight feature weighting module based on YOLO9000 (YOLOv2) [6], enabling the detector to quickly adapt to new categories. The feature learner is trained with sufficient samples from base class datasets to extract meta-features that can be generalized to new object classes. The weighting module converts some supporting examples from the new class into global vectors, indicating the importance or correlation of the meta-features of the corresponding detected object. By integrating the meta-features learned by the feature learner and the weight vector convolution obtained by the weighted module, the classification and regression information of the object can be obtained. However, these networks are too limited to extract features so that many features are abandoned when extracting features. In addition, the network is just weighted by a simple channel convolution, failing to fully use of the existing new object classes information.

For these remaining problems, we propose the THR network in order to extract more features of a given picture and to use the support set to perform feature fusion, so as to detect the object. This THR method learns the meta feature of the general object from the base class and then transfers the learned meta feature to detect other objects. Specifically, the model trained on a large amount of label data can extract discriminative features, which is also effective for some new categories. The proposed method needs generalization ability. When using a few samples of the new class to fine-tune some of the parameters in the network, the network can detect the object of the new class, which is equivalent to the knowledge learned in the base class migrating to a new category.

The underlying characteristics of the base class can reflect the commonality of all classes (base class + new class), and can also be understood as the basic attributes of the constituent objects because there are still many similarities at the attribute level. For example, common cats and dogs have four approximately cylindrical legs, the same to the new categories of pigs, cows, and sheep, which also indicates the similarities between few-shot learning and transfer learning. In summary, our network requires two-stage training. First, it is trained with a large amount of base class data without the new class, and then the training is fine-tuned with the new class.

In general, the major improvements of this article are summarized below.

1) We propose a THR network based on high-resolution representation and transformer for feature fusion, the first feature fusion network that uses the transformer as query sets and support sets in few shot object detection.
2) In the backbone network, we use the High-resolution network as the feature extraction network and add parallel spatial and channel attention modules for each multi-layer feature.
3) In this fusion module, based on the of the transformer, we simplify and modify the transformer structure, and finally get a feature enhancement and fusion module. The query set and support set features are enhanced through two parallel weights sharing self-attention in the feature enhancement. Then the fusion module is used to fuse the obtained vectors.

## 2 Related Works

### *2.1 Object Detection*

Deep learning object detection is developed on image classification [7–14]. Object detection includes object classification and object location. The existing methods of object detection mainly conclude one-stage [15,16] and two-stage object detection [17,18], according to whether candidate boxes are needed or not. The two-stage method needs to extract candidate boxes with potential objects first, then classify and evaluate the bounding boxes of the extracted candidate boxes [19]. The one-stage method uses the anchor method to decide whether each grid is on the center point of the object, and finally perform regression and classification of the detection of the object without using the candidate boxes. Generally speaking, the two-stage method has better detection accuracy than the one-stage method [20], while the one-stage method weighted more than two-stage method in the aspect of detection speed. Generally, both one-stage and two-stage methods can achieve good results in object detection. However, these methods have a common disadvantage that they all need sufficient data and labels to train the network. If the object data are relatively limited, the detection results might not be satisfactory due to the data imbalance and overfitting.

### *2.2 Few Shot Object Detection*

Few shot learning is the first method proposed to train image classification with a few data [21,22]. The few shot object detection, including the few-shot classification and objects location, is to detect objects with only a small amount of training sample. Usually, we can divide it into methods based on meta-learning and methods based on fine-tuning [23].

The method based on fine-tuning mainly means that serving the detection model obtained from a large number of similar data sets as the initial model, which could be adapted to new detection applications only after optimizations. Singh et al. [24] introduced a pre-training model to improve the accuracy of the model in object detection with sparse samples. Chen et al. [25] selected object knowledge from the source domain and object domain respectively when doing domain adaptation

to further enhance the fine-tuning of few shot object images. The method of using a pre-training model and fine-tuning is extensively employed in the object detection field. After fine-tuning, both the classification and location capabilities of the model will be improved. In addition, the model convergence time could be greatly reduced by the introduction of the pre-training model. However, the pre-training model also brings about model overfitting, which needs regularization method to alleviated [26]. Besides, although pre-training and fine-tuning can converge on the training set, the detector still fail to solve the problem of insufficient generalization ability on the test set.

Meta-learning aims to let the model learn to learn. Meta-learning fits the distribution of a series of similar tasks, and synthesizes the parameters of each learning task through the meta-learner in order to obtain a good initialization parameter. Because of the advantages of meta-learning, increasing researchers have focused on employing meta-learning in their studies. First, the model proposed by Fu et al. [27], combining the meta-learning part and the object detector, could learn a wide range of knowledge and correct rapid adaptation strategies, and teach the detector to learn from a limited number of examples. Moreover, the MetaDet proposed by Yan et al. [5] employed meta-learning based on region of interests (ROI) characteristics and improves the object segmentation ability. Besides, Wang et al. [28] have unique ideas, believing that the combination of object detection and few-shot learning is a special case of the object tracking problem. With the adoption of model-independent meta-learning, they provided a strategy for initializing the detector, built a high-performance tracker and adopted a series of optimization schemes for few samples, which greatly improves the detection accuracy. Pérez-Rúa et al. [29] added a meta-learning category generator to the network. Once trained, given a small number of images of new object classes, the meta-training generator can help the once detector learn new classes with an effective feed-forward manner in the meta-testing stage. Compared with the YOLO model, Kang et al. [3] fully applied the label of the base class, along with the meta feature learner and weight adjustment module, and finally achieved rapid adaptation to the new class, which possesses great advantages in the object detection with a few samples. Therefore, this approach suggests feasibility for our designed THR network.

### *2.3 Transformer*

Transformer, first used in natural language processing (NLP), employs an encoder-decoder architecture [30]. The architecture first converts the input features into three matrices: Query, key, and value, then uses the dot product between query and key to obtain the attention weight of each input word. This process, also the core of the Transformer, is called self-attention calculation. With the use of self-attention weights, the features would be conducted weighted summation, and would find that there are feature outputs with special associations between words. Transformer and its variants have achieved excellent results in natural language processing. The most famous one is the pre-training of deep bidirectional transformers (BERT) [31], which pre-trains translators for unlabeled text by jointly limiting the contexts.

Considering the advantages of the self-attention mechanism [32], researchers are engaging in applying transformers to computer vision. In the past, scholars are committed to improving the performance of computer vision in convolutional neural network (CNN). Due to the emergence of the transformer, researchers have discovered that the transformer can become an important substitute for CNN in computer vision. In this aspect, end-to-end object detection with transformers (DETR) is the first network that applies transformer to object detection [33]. Later, vision transformer (VIT) recently proposed that computer vision problems could be solved by a pure transformer and achieved state-of-the-art (STOA) performance on multiple image recognition benchmarks [34]. Due to the satisfactory performance of the transformer, increasing computer vision models based on the transformer have

been proposed, which sheds new light on our research. For this potential advantages of transformer in the field of few shot object detection, we take some deletions and modifications to the structure of the generic transformer in order to make this process more efficiently.

## 3 The Proposed Approach

### *3.1 Problem Definition*

In this network, we provide base class data with sufficient data and new class data with only a small amount of data. Our goal is to use base class and new class data to train a network that can detect both base objects and new class objects. As Fig. 2 shows, we performed two-step training. First, we train the prior knowledge that can generalize image features on the base class, and then help the network quickly adapt to the detection of the new class through fine-tuning from the training of the new class. In this study, we define the data of the base class as $B$, and define the data of the new class as $R$. Our network always has two-way data inputs from the training of the base class and the new class in which one-way input is the support set data $s$, and the other data input is the query set, which is defined as $q$. In the base class training, we set the data of the support set as $B_s$, and the data of the query set as $B_q$, which is also represented in the new class. If the number of categories in the new category is $N$, and the number of images in each category is $k$, the problem is defined as $N$-way $k$-shot detection.

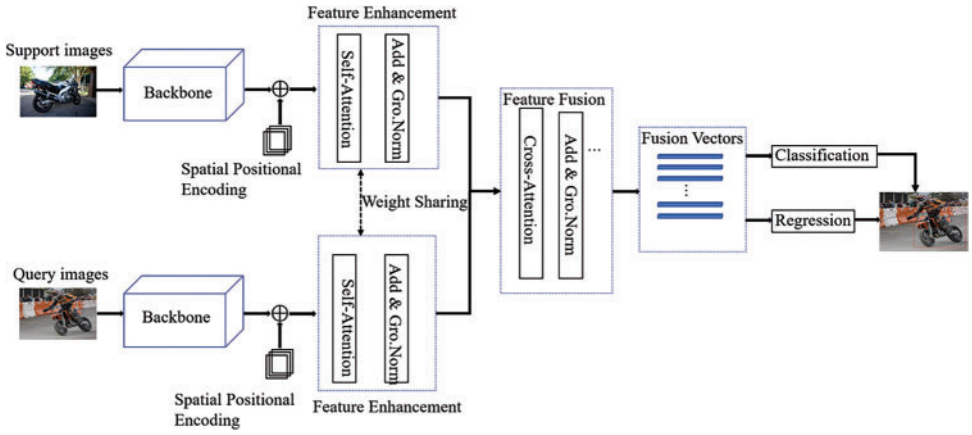


**Figure 2:** This is our proposed THR network framework. It contains two branches, a query branch, and a support branch. It is mainly composed of three modules: Backbone extraction network, feature enhancement and feature fusion, and classification and regression module

### *3.2 Network Overview*

The network proposed mainly contains a feature extraction module with high-resolution representation, which is used to extract features from the support set and query set. After that, the extracted features would be flattened into feature vectors, then the feature vectors would be inputted into the feature enhancement and feature fusion of the modified transformer to fuse the feature vectors from the query set and the support set, finally achieving the classify and location of the obtained fusion vector.

### *3.3 Network Overview*

The network proposed mainly contains a feature extraction module with high-resolution representation, which is used to extract features from the support set and query set; After the extracted features are flattened into feature vectors, they are input into the modified transformer encoding and decoding network to fuse the feature vectors obtained from the query set and the support set; Finally, classify and locate the obtained fusion vector.

### *3.4 High-resolution Feature Extraction Network*

To extract the high-resolution feature and lose the features of the image as little as possible, this paper uses a high-resolution feature network. This detailed feature extraction is shown in Fig. 3 below.

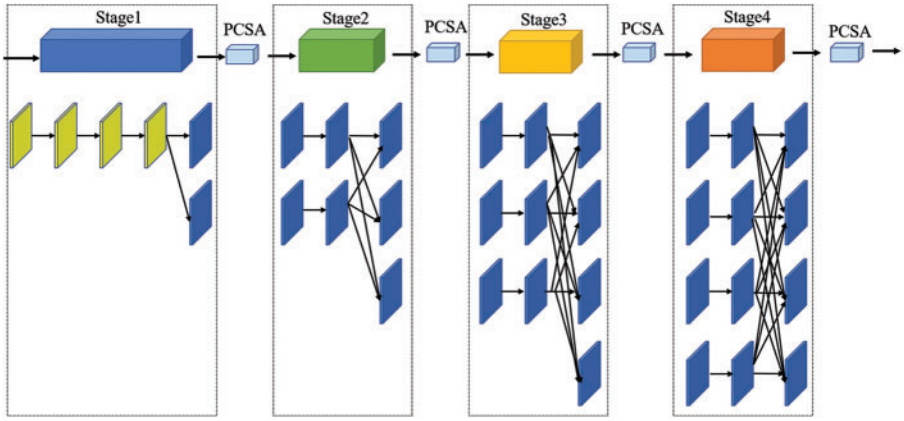


**Figure 3:** Structure of feature extraction network. It consists of four stages, and each stage is followed by one more branch after being fused through the stride convolution. After each stage, there is a channel and spatial attention module

#### *3.4.1 Overview of the Backbone Network*

With the consultant of the network by sun et al. [35], we modified our network as well. This network is divided into four stages in which each stage has one more branch than the previous stage. The new branch conducted stride convolution fusion of all the feature maps of the previous stage. Therefore, the resolution of the new one is half of the resolution of the previous branch, while the number of channels is doubled. Each stage performs feature extraction first and then performs multi-scale fusion, specifically, using stride convolution for high-resolution images, and using up-sampling, and $1 \times 1$ convolution for low-resolution images. Then the feature maps of three different resolutions of the high, medium, and low could be merged mutually. Because the fusion strategy is pixel addition, the number of channels with different resolution feature maps needs to be adjusted to the same. In addition, to increase the receptive field, we use hole convolution to extract features in the first stage of convolution. At the same time, in order to enhance the efficiency of accuracy of the feature map, a parallel channel and spatial attention at the end of each stage are applied.

#### *3.4.2 Parallel Channel and Spatial Attention Module (PCSA)*

Comparing to the attention model in Bottleneck attention module (BAM) [36], we added the PCSA module to the network, which could be seen in Fig. 4. The PCSA module, containing parallel

channels and spatial attention, is connected through a similar structure with the residual network. For BAM, two fully connected layers are used in the channel attention. Different from BAM, the channel attention in PCSA uses two $1 \times 1$ convolution and relu activation functions in which the nonlinear expression ability of the functions is improved. Besides, BAM applies hole convolution, while this spatial attention of PCSA uses depthwise separable convolution to filter out some information, and can reduce the complexity and parameters of the module. Among them, Channel attention module (CAM) focuses channels attention and determines which channels has the main characteristics of the object. However, spatial attention module (SAM) focuses on the spatial location and determines which location contains the main information of the object. Therefore, our proposed module needs less increase of the parameters of the model, which is not only less complex, but also increase its attention at the same time. The PCSA is shown in Eq. (1).

$$F_{PCSA} = x_C \cdot \sigma \left(F_{SAM} + F_{CAM}\right) \tag{1}$$

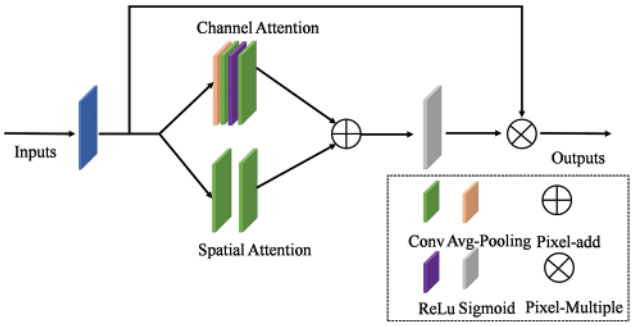


**Figure 4:** Parallel channel and spatial attention module

Eqs. (2) and (3) represent $F_{SAM}$ and $F_{CAM}$, respectively.

$$F_{SAM} = W_P \left(W_G \left(x_c\right)\right) \tag{2}$$

$$F_{CAM} = w_u \left(\delta \left(W_D \left(F_{GAP} \left(x_c\right)\right)\right)\right) \tag{3}$$

$F_{PCSA}$ represents the feature map of the processed output and $x_C$ indicates the feature map of the input. Activation function $\sigma$ and $\delta$ represent sigmoid and relu function respectively. $W_P$, $W_D$, and $w_u$ represent the convolution operations. $F_{GAP}$ represents maximum average pooling. $F_{SAM}$ is the output of spatial attention and $F_{CAM}$ is the output of channel attention.

Compared with the attention module for solving the classification of convolutional block attention module (CBAM) [37], this paper uses a parallel template design to learn complex image features, which is more suitable. Through parallel channels and spatial attention modules, we can learn the correlation of channel characteristics at each location and adjust them accordingly, which can effectively enhance the expressive ability of features. In another word, PCSA not only improves the performance of the network but also reduce the computational load as much as possible.

### *3.5 Feature Enhancement and Feature Fusion*

#### *3.5.1 Overview of Feature Enhancement and Feature Fusion*

The Transformer [30] network was originally proposed by Google in 2017. THR uses an attention mechanism to extract features in parallel from the input data. To maintain data correlation, the network uses position-coding to record the position information. Therefore, on the one hand, the

structure can still ensure the correlation between the data before and after; on the other hand, due to the parallel input, the network training time is greatly shortened. Figs. 5 and 6 show the basic structural unit of the feature enhancement and feature fusion network. During feature extraction, the input data is first sent to the feature enhancement module for data enhancement, and the autocorrelation and other features of the data are obtained. Then the two features are fused by the feature fusion module to focus on the edge and similar target features.

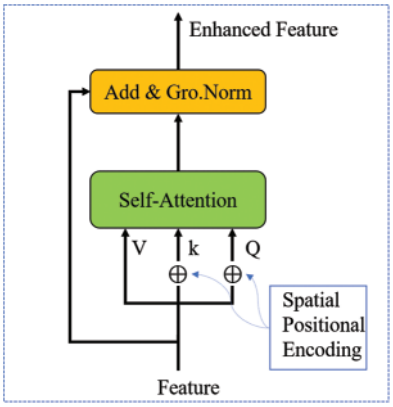


**Figure 5:** Feature enhancement structure. This structure is to enhance query set feature vectors and support set feature vectors

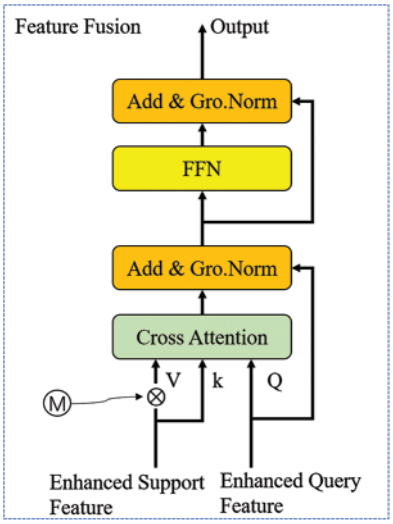


**Figure 6:** Feature fusion structure. It is used for feature fusion of query set and support set output by the Feature Enhancement

The feature enhancement and feature fusion module is derived from transformer. However, this module in our research is very different from transformer and is more suitable for few shot object detection. The main differences are as follows: Although these two modules both use the attention mechanism, but cross attention is used in this paper and ordinary attention is used in transformer. In addition, group normalization is used in our study, while conventional normalization is used by

transformer. Finally, compare to the the complex structure in transformer, our structure is more concise, which could decrease the complexity of the transformer model.

### *3.5.2 Feature Enhancement*

Feature enhancement is to enhance the support feature and query feature. These two features are enhanced separately in the same way. Before calculating self-attention, three input vectors were created by positional encoding, which are defined as Q, K, and V. After getting Q, K, V, we first calculate the value of attention, and the steps are as follows:

1) Calculate the correlation score in the input vector, which is calculated through the dot multiplication method, calculating the dot multiplication with each vector in $Q$ and each vector in $K$. As shown in Eq. (4).

$$score = Q.K^T \tag{4}$$

2) Normalize the calculated correlation score. The main purpose of normalization is to stabilize the gradient during training. See Eq. (5).

$$score = score\ /\tau \tag{5}$$

Among them, $\tau$, a parameter that controls the distribution of softmax, refers to model disintegration and contrastive learning technologies.

3) Convert the normalized score vector into a probability distribution between [0–1] through the softmax function. After softmax, the score is transformed into a probability distribution matrix A with values distributed between [0, 1]. As shown in Eq. (6).

$$A = Atten\,(Q, K) = Softmax_{col}\left(QK^T/\tau\right) \tag{6}$$

4) Multiply the corresponding value according to the probability distribution, that is the dot product of A and V. See Eq. (7).

$$Z = A \cdot V \tag{7}$$

The "Add" in Fig. 5 means to add a residual block. The purpose of adding residual block is to prevent degradation problems in deep neural network training. Degradation means the dynamic relationship between the loss of network and the increase of the number of layers in the deep neural network. In particular, the loss of network would be gradually decreased at the beginning if the number of layers increased, then the loss of network would be gradually stabilized, even reaching saturation, however, the loss would increase instead in the end with the continuing increase of the number of network layers.

The Normalization of the input data possesses two purposes. The first is to speed up training and the other is to improve the stability of training. However, batch normalization will bring some problems. Due to the close relationship between Batch normalization (BN) layer and the batch size, the so if the batch size is too small, the effect of BN would receive too much interference. Because each calculation of mean and variance is on a batch, this result can not fully reflect the distribution of the whole data, which can easily lead to too much memory and too long training time. In addition, the training may not be successful because of the fixed gradient descent direction. Comparing to the deficiency of BN, group normalization is more effective. Group normalization (GN) divides the channel into several groups and calculates within the group. The calculation of GN has nothing to do with the batch size, and the accuracy of the calculation is stable in the large batch range. For images

with input sizes of [N, C, H, W], GN first sets channels into groups, then calculates variance and mean in each group, helping the input of each layer obey the normal 0–1 distribution, which solves the problem of intermediate covariance migration and accelerates the convergence of the network. GN is Eq. (8).

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta \tag{8}$$

Among them, $x$ is the input and $y$ is the normalized result. $E[x]$ and Var$[x]$ are the mean and standard deviation respectively. $\gamma$ and $\beta$ are learnable parameters, and $\epsilon$ here is a small value close to 0 which is added to prevent the denominator from being 0.

#### *3.5.3 Feature Fusion*

In order to transfer the support set feature to the query set feature, we use the obtained vectors as the input $Q$ and $K$ respectively, which are gained from the feature enhancement of query set and the support. At the same time, in order to suppress the background, the area outside the support concentration object, we also take the label mask of the enhanced support set vectors as the input in which the encoded support set vectors could be obtained through S $\otimes$ M. After that we calculate the changed features through cross-attention $A_{S\to Q}(S \otimes M)$. See Eq. (9).

$$Q_{feat} = Gro.Norm\left(A_{S\to\hat{Q}}(S \otimes M\ ,\ K, Q) + Q\right) \tag{9}$$

The feature-level enhanced $Q_{feat}$ aggregates different object representations in time from a series of query features $Q$.

After that, the fused features are inputted into a feed-forward network (FFN) with skip connection. Then the extracted feature vectors from cross-attention would conduct FC feature conversion, which could increase the expressive ability of the model. The FFN is a two-layer Multi-Layer Perceptron (MLP), which is composed of a fully connected layer and a nonlinear activation function, which can be applied to each position separately. As shown in Eq. (10).

$$FFN(H') = ReLU\left(H'W^1 + b^1\right)W^2 + b^2 \tag{10}$$

Here $H'$ is the output of the previous layer. $W^1 \in \mathrm{R}^{D_m \times D_f}$, $\mathrm{W}^2 \in \mathrm{R}^{D_f \times D_m}$, $\mathrm{b}^1 \in \mathrm{R}^{D_f}$, $\mathrm{b}^2 \in \mathrm{R}^{Dm}$ are all trainable parameters. According to experience, the value of $D_f$ is usually larger than $D_m$. After passing FFN, we then used the same add and group norm modules in 3.4.2.

## 4 Experiments

In this part, we mainly compare and evaluate the proposed model THR through experiments. In this paper, the combination of transformer and CNN is mainly used to detect few shot object. The details are as follows.

### *4.1 Experimental Setting*

#### *4.1.1 Datasets*

We use general object detection data sets to train and evaluate our network. The used data sets are mainly Pascal VOC2007, Pascal VOC2012, and MS COCO. The settings of data sets of the THR are the same with the FSODFR [3].

#### *4.1.2 PASCAL VOC*

We use the VOC2007 + 12 datasets to train on the train & val set of VOC2007 and VOC2012 (16551 pictures), and apply the test set of VOC2007 (4952 pictures) to test. For the training set, select most categories as base classes and the remaining categories as new class in which Base classes contain a large number of tagged image data, and Novel class only contains a few images. For the N-way K-shot detection problem, we set the Novel class to N categories, and each category has K pictures with labels. First, we perform basic training on the base classes to get an initial network weight and then fine-tune the network on the novel class. In the novel class, we will also add the object in the base class so that the trained network can detect both the new class and the base class. To prevent the particularity of network detection, we divide the VOC three times to train and detect the model. In each division, for the 20-category Pascal VOC data set, five categories are randomly selected as the new category, and the remaining 15 categories are used as the base category data. For each division, we take 1, 2, 3, 5, and 10 for the K value of the new class for training and testing. When evaluating the VOC data set, we use the mean average precision of the new class to evaluate. When the intersection and union ratio between the detection result and the label is greater than 0.5, it is correct. This way of evaluation is called AP50.

#### *4.1.3 MS COCO*

The COCO data set contains rich categories and massive pictures. It can be used for research in various directions, including image title generation, object detection, keypoint detection, and instance segmentation. For the object detection task, MS COCO contains a total of 80 categories in which the training set and the validation set contain more than 120,000 pictures and more than 40,000 test pictures together. For the MS COCO data set, because of its contained 80 categories, we choose its 20 categories as the new category and the remaining 60 categories as the base category.

#### *4.1.4 Training Details*

The experimental environment is two RTX2080Ti GPUs with 32GB of memory. Our research is implemented through python programming based on the Linux platform and uses the deep learning framework PyTorch to build and reproduce all network models. Network parameters are setted accordingly in which stochastic gradient descent is used in gradient descent, momentum is set to 0.9, weight attenuation is set to 0.0003, and batch size is set to 16. At the same time, the training images are processed to enhance and expand the training data set by methods such as horizontal flipping, vertical flipping, color dithering, and exposure adjustment [38].

### *4.2 Comparison of Experimental Results*

#### *4.2.1 Results on Pascal VOC*

In Tab. 1, we can see the test results of the network model trained on the Pascal VOC of the THR network on the new class. At the same time, we compare its results with other existing one-stage models, including A Low-Shot Transfer Detector (LSTD) [25], FSODFR, and MetaDet [5]. The THR detector we proposed can have a better detection effect when the shot of the new class is large. Specifically, in the first data set division, we improved 1.2 points over the best in 3 shots, 2.7 points over the best in 5 shots, and 1.1 points over the best in 10 shots. However, we can also find that when the shot is 1 or 2, our experimental results are not as good as the previous best results. From our analysis, the possible reason could be the relatively small shot, then the limited new class samples could not well applied to new classes during feature transfer. Although some experimental results are not as good as

the previous research, our average improvements in the accuracy of the few shot object detection in the first, second and third divisions in mean average precision (mAP) are 0.66, 2.28, and 2.14 respectively.

**Table 1:** On VOC, divide the data set randomly three times, take 1, 2, 3, 5, and 10 for the divided data sets shot respectively, and get the AP50 result

| | Novel set1 | | | | | Novel set2 | | | | | Novel set3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Shots | 1 | 2 | 3 | 5 | 10 | 1 | 2 | 3 | 5 | 10 | 1 | 2 | 3 | 5 | 10 |
| LSTD | 8.2 | 11.0 | 12.4 | 29.1 | 38.5 | 11.4 | 3.8 | 5.0 | 15.7 | 31.0 | 12.6 | 8.5 | 15.0 | 27.3 | 36.3 |
| YOLOv2-ft | 6.0 | 10.7 | 12.5 | 24.8 | 38.6 | 12.5 | 4.2 | 11.6 | 16.1 | 33.9 | 13.0 | 15.9 | 15.0 | 32.2 | 38.4 |
| FSODFR | 14.8 | 15.5 | 26.7 | 33.9 | 47.2 | 15.7 | 15.3 | 22.7 | 30.1 | 40.5 | 21.3 | 25.6 | 28.4 | 42.8 | 45.9 |
| MetaDet | 17.1 | 19.1 | 28.9 | 35.0 | 48.8 | 18.2 | 20.6 | 25.9 | 30.6 | 41.5 | 20.1 | 22.3 | 27.9 | 41.9 | 42.9 |
| Our | 16.2 | 19.0 | 29.4 | 37.7 | 49.9 | 17.2 | 18.9 | 27.8 | 36.8 | 47.5 | 20.0 | 23.5 | 28.9 | 44.2 | 49.2 |

#### *4.2.2 Results on MS COCO*

Compared with VOC data sets, COCO datasets show greater complexity in object detection, because COCO data has 80 categories and more pictures. We train the network with 60 base classes of COCO, and then fine-tune the network when the shot is 10 or 30 respectively. The experimental results are shown in Tab. 2. Through comparison, our experimental results are better than the previous methods. For shot is 10, our method increases by 2.7 points at AP50:95, and for shot is 30, our method increases by 2.8 points at AP50:95.

**Table 2:** On the COCO data set, the results of this THR method and other methods in few shot object detection when the shot is 10 and 20 respectively

| | | Avg.precision | | |
|---|---|---|---|---|
| Shot | Method | 0.5:0.95 | 0.5 | 0.75 |
| K = 10 | LSTD | 3.2 | 8.1 | 2.1 |
| | FSODFR | 5.6 | 12.3 | 4.6 |
| | MetaDet | 7.1 | 14.6 | 6.1 |
| | Our(THR) | 9.8 | 17.9 | 9.7 |
| K = 30 | LSTD | 6.7 | 15.8 | 5.1 |
| | FSODFR | 9.1 | 19.0 | 7.6 |
| | MetaDet | 11.3 | 21.7 | 8.1 |
| | Our(THR) | 14.1 | 25.6 | 14.2 |

### *4.3 Ablation Study*

To test the hole convolution and spatial and channel attention in high-resolution networks, we added ablation experiments of these modules. We conducted experiments in a conventional high-resolution network. As shown in Tab. 3, We found that the experimental results were improved when

hole convolution or spatial and channel attention were added. When both were added to the network, the results would be better improved. Therefore, we believe that increasing the receptive field of the model and giving some weight to the feature map in the model through the attention mechanism are effective.

**Table 3:** The ablation experimental results when the feature extraction network is added with hole convolution, channel and spatial attention

| Hole convolution | Spatial and channel attention | AP50 |
|---|---|---|
| | | 0.3345 |
| √ | | 0.3452 |
| | √ | 0.3466 |
| √ | √ | 0.3548 |

Tab. 4 shows that when we use cross attention instead of self-attention, we can better fuse the support set feature vectors and the training set vectors. In addition, when mask is used to replace the previous spatial positional coding in the Feature fusion module, the detection results of the network can also be improved.

**Table 4:** The ablation experimental results when the mask and cross attention are added to feature enhancement and feature fusion module

| Mask | Cross attention | AP50 |
|---|---|---|
| | | 0.3005 |
| √ | | 0.3220 |
| | √ | 0.3342 |
| √ | √ | 0.3432 |

After the above Ablation Experiment on the feature extraction network and feature enhancement & feature fusion module, we select the best combination of the two methods to do the Ablation Experiment on the high-resolution feature extraction network, feature enhancement &feature fusion module and two training in the network. Tab. 5 shows the test results after training. We also verified the role of the added modules and our training methods. Through the experimental results, we could see the fine-tuning method can greatly improve object detection.

**Table 5:** Ablation experiments of high-resolution feature extraction network, feature enhancement and feature fusion and fine-tuning used in THR network

| Fine-tuning | High-resolution network | Feature enhancement & feature fusion | AP50 |
|---|---|---|---|
| | | | 0.1547 |
| √ | | | 0.3202 |

(Continued)

**Table 5:** Continued

| Fine-tuning | High-resolution network | Feature enhancement & feature fusion | AP50 |
|---|---|---|---|
| √ | √ | | 0.3548 |
| √ | | √ | 0.3432 |
| √ | √ | √ | 0.3770 |

## 5 Conclusion

In this paper, we propose a THR network to realize few shot object detection. In the network, we use a high-resolution feature extraction network to extract network features, and use cross attention transformer to fuse query set features and support set features. Therefore, a simple and effective detector is constructed by combining high-resolution network and transformer. Experiments show that our proposed THR network is better in the accuracy of object detection than the previous networks when the shot is greater than 3, which could solve some of the remaining problems in this area. However, for the increasing needs of more accuracy in detection, there are still some deficiencies so that we will use various methods to improve its detection accuracy in the future.

**Funding Statement:** This project is supported by the National Natural Science Foundation of China under grant 62172059 and 62072055, Hunan Provincial Natural Science Foundations of China under Grant 2020JJ4626, Scientific Research Fund of Hunan Provincial Education Department of China under Grant 19B004, "Double First-class" International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology under Grant 2018IC25, and the Young Teacher Growth Plan Project of Changsha University of Science and Technology under Grant 2019QJCZ076.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580–587, 2014.

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779–788, 2016.

[3] B. Y. Kang, Z. Liu, X. Wang, F. Yu, J. S. Feng *et al.,* "Few-shot object detection via feature reweighting," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Seoul, Korea, pp. 8420–8429, 2019.

[4] Y. X. Wang, D. Ramanan and M. Hebert, "Meta-learning to detect rare objects," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Seoul, Korea, pp. 9925–9934, 2019.

[5] X. Yan, Z. L. Chen, A. N. Xu, X. X. Wang, X. D. Liang *et al.,* "Meta R-CNN: Towards general solver for instance-level low-shot learning," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Seoul, Korea, pp. 9577–9586, 2019.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 7263–7271, 2017.

[7] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 37, no. 9, pp. 1097–1105, 2012.

[8] M. Lin, Q. Chen and S. C. Yan, "Network in network," in arXiv preprint arXiv:1312.4400, pp. 1–10, 2013.
[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in arXiv preprint arXiv:1409.1556, pp. 1–14, 2014.
[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, Lille, France, pp. 448–456, 2015.
[11] C. Szegedy, W. liu, Y. Q. jia, P. Sermanet, S. Reed *et al.,* "Going deeper with convolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 1–9, 2015.
[12] K. M. He, X. Zhang, S. Q. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770–778, 2016.
[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2818–2826, 2016.
[14] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI Conf. on Artificial Intelligence*, San Francisco, CA, USA, pp. 4278–4284, 2017.
[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 1804–2767, 2018.
[16] A. Bochkovskiy, C. Y. Wang and H. -Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.
[17] R. Girshick, "Fast R-CNN," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Boston, MA, USA, pp. 1440–1448, 2015.
[18] S. Q. Ren, K. M. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
[19] D. Y. Zhang, J. W. Hu, F. Li, X. L. Ding, A. K. Sangaiah *et al.,* "Small object detection via precise region-based fully convolutional networks," *Computers, Materials & Continua*, vol. 69, no. 2, pp. 1503–1517, 2021.
[20] S. Samanta, M. Panda, S. Ramasubbareddy, S. Sankar and D. Burgos, "Spatial-resolution independent object detection framework for aerial imagery," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 1937–1948, 2021.
[21] Y. B. Chen, X. L. Wang, Z. Liu, H. J. Xu and T. Darrell, "A new meta-baseline for few-shot learning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 880–890, 2020.
[22] B. Zhang, H. Ling, P. Li, Q. Wang, Y. Shi *et al.,* "Multi-head attention graph network for few shot learning," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 1505–1517, 2021.
[23] J. Lu, P. H. Gong, J. P. Ye and C. S. Zhang, "Learning from very few samples: A survey," in arXiv preprint arXiv:2009.02653, pp. 1–30, 2020.
[24] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection snip," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Salt Lake City, UT, USA, pp. 3578–3587, 2018.
[25] H. Chen, Y. L. Wang, G. Y. Wang and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," in *Proc. of the AAAI Conf. on Artificial Intelligence*, New Orleans, LA, USA, pp. 2836–2843, 2018.
[26] Y. Q. Wang, Q. M. Yao, J. T. Kwok and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.
[27] K. Fu, T. F. Zhang, M. L. Yan, Z. H. Chang, Z. Y. Zhang *et al.,* "Meta-SSD: Towards fast adaptation for few-shot object detection with meta-learning," *IEEE Access*, vol. 7, no. 6, pp. 77597–77606, 2019.
[28] G. T. Wang, C. Luo, X. Y. Sun, Z. W. Xiong and W. J. Zeng, "Tracking by instance detection: A meta-learning approach," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 6288–6297, 2020.
[29] J. -M. Perez-Rua, X. Zhu, T. M. Hospedales and T. Xiang, "Incremental few-shot object detection," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 13846–13855, 2020.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," *Advances in Neural Information Processing Systems*, Los Angeles, CA, USA, pp. 5998–6008, 2017.
[31] J. Devlin, M. -W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in arXiv preprint arXiv:1810.04805, pp. 1–16, 2018.
[32] H. P. Wu, Y. L. Liu and J. W. Wang, "Review of text classification methods on deep learning," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1309–1321, 2020.
[33] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov *et al.,* "End-to-end object detection with transformers," in *European Conf. on Computer Vision*, Glasgow, US, pp. 213–229, 2020.
[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai *et al.,* "An image is worth 16x16 words: Transformers for image recognition at scale," in arXiv preprint arXiv:2010.11929, pp. 1–22, 2020.
[35] K. Sun, B. Xiao, D. Liu and J. D. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 5693–5703, 2019.
[36] J. Park, S. Woo, J. -Y. Lee and I. S. Kweon, "Bam: Bottleneck attention module," in arXiv preprint arXiv:1807.06514, pp. 1–14, 2018.
[37] S. Woo, J. Park, J. Y. Lee and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. of the European Conf. on Computer Vision, Munich*, BAV, GER, pp. 3–19, 2018.
[38] X. Shen, G. Dong, Y. Zheng and L. Tsang. "Deep co-image-label hashing for multi-label image retrieval," *IEEE Transactions on Multimedia*, vol. 24, no. 1, pp. 1116–1126, 2021.