Mixed Feature Processing Model for Few-Shot Object Detection

Qian Jiang$^{1[0000-0003-3097-0721]}$, Shuting Li^{1} ,Dakai Sun^{1} , Yu Wang^{2} , Biaohua Liu^{1} ,Shengfa Miao^{1} and Xin Jin$^{1[0000-0003-2211-2006]*}$

[1] School of Software, Yunnan University, Kunming 659000, China
[2] Information Technology Department, The Second Affiliated Hospital of Kunming Medical University, Kunming 659000, China
[*] Corresponding Author: xinxin_jin@163.com

Abstract. Traditional object detection methods typically require large-scale annotated training data. However, in some areas, acquiring a large amount of annotated data can be extremely challenging. To address the issue of Few-Shot Object Detection (FSOD), researchers have introduced the concept of meta-learning. Currently, meta-learning is widely applied in two-stage object detection. We have identified several key issues affecting the accuracy of FSOD, including limited data, insufficient feature extraction capabilities, and the aggregation method between different features. To more finely extract features and better aggregate features, we separate the support branch and query branch of Meta-RCNN, forming two parallel branches. We create one mixed feature processing model for few shot object detection. We put the Feature Pyramid Network (FPN) only into the backbone network of the query branch, creating a strong baseline to enhance the extraction capabilities for images of different dimensions. Additionally, for the first time in FSOD. We use a Variational Autoencoder (VAE) model to extract features, which achieves data augmentation and improves the generalization ability of the network by adding the VAE to the support branch to obtain more useful information in the support set. In addition to this, we design a module *R* to aggregate the output support image features with the query image features on the query branch. The aggregated results are fed into the detection head of the object detection process. Experimental results demonstrate that the proposed method exhibits good performance. Following the experimental settings for FSOD, we conducted extensive experiments on the PASCAL VOC dataset, showing that our method is superior to other methods currently available and achieves very satisfactory results.

Keywords: Few-Shot Object Detection, Feature Pyramid Network, Variational Autoencoder.

1 Introduction

With the continuous development in the field of computer vision, research based on images is gradually progressing [1][2]. Traditional object detection based on deep learning has achieved good performance. Object detection models require many images and annotations, such as bounding boxes and class labels. Due to the detection performance depends heavily on the number of images and annotations, it is difficult for the model to achieve good training results in case of insufficient data. The core challenge of FSOD is to learn and accomplish the task of object detection on a limited dataset. Therefore, various methods have been proposed to address the FSOD problem, including meta-learning, metric learning, and fine-tuning. In this paper, we will specifically focus on meta-learning-based methods, using Meta R-CNN[3] as the baseline network for exploring FSOD. We found that the ultimate challenge of these methods is how to extract more information from the dataset and fully utilize the useful information.

Most FSOD algorithms based on meta-learning are built upon Faster-RCNN [4], where the feature maps pass through the Region Proposal Network (RPN) after the backbone network and are directly fed into the detection head via pooling layers. Since the object sizes in Pascal VOC images vary slightly, using only the C4 layer of the backbone network can yield satisfactory results. However, in datasets where objects vary significantly in size, the network's performance is affected, resulting in many instances of false positives and false negatives. To better address the issue of small object detection in FSOD, we introduce the combination of FPN with existing meta-learning methods, which serves as a strong baseline. Compared to methods solely relying on the C4 layer, this enhancement can better handle datasets with large scale variations. FPN works by constructing FPN to handle feature information at different scales, improving the adaptability of the object detection model to scale variations. The main idea of FPN is to establish a top-down feature pyramid structure in deep neural networks, allowing the model to simultaneously acquire features from both low-level and high-level layers, thus better capturing the features of objects at different scales.

Meanwhile, due to the lack of data, the model is prone to overfitting, and we believe that the simplest way to alleviate this problem is to annotate the images, but it is inefficient in terms of time and cost. Therefore, we consider using data augmentation methods, which mainly include image processing methods and deep learning methods. In previous studies, researchers used GANs [5-8] for data enhancement. GANs can generate real images that are closer to the original, whereas VAEs are more concerned with the well-structured latent space of the image. As a result, VAE[9] models are often used in the field of image generation compared to GANs. In FSOD, it is necessary to extract more hidden features from the dataset to enrich the information passed to the detection head. Therefore, in this paper, we chose to incorporate the VAE model into the support branch of the meta-learning model so that the network can learn more features from the support set. VAE is a generative model that focuses on learning the latent distribution of data. In FSOD tasks, due to the limited number of samples, learned feature representations become particularly crucial. Its structure consists of an encoder and a decoder, which can learn the latent representations of support set images, capturing key features Bin the images and helping to improve the model's generalization ability in Few-Shot scenarios. Additionally, the decoder part of VAE can be used to generate new samples with a similar distribution to the support set images. This effectively performs data augmentation by generating variants of support set samples during the training process, thereby effectively expanding the training data and enhancing the model's robustness to various changes. Data augmentation is crucial in FSOD because limited samples may not cover various scenarios and variations.

To evaluate our model, we conducted experiments on the PASCAL VOC dataset to validate the effectiveness of our approach. Our main contributions are:

- We created a strong baseline network by incorporating the FPN only into the query branch to adapt to datasets with large scale variations.

- We first introduced the VAE model is added to the support branch of meta-learning FSOD and the VAE model is used to extract more image information to mine out the hidden features from the data images.

- We created three kinds of different models to aggregate the output results, and ultimately opted for the simplest one, which uses a fully connected layer to aggregate features and feed them into the detection head.

2 Related work

2.1 Generic Object Detection

Traditional object detectors are typically categorized into single-stage and two-stage detectors. Single-stage detectors like the YOLO series[10]-[13] utilize backbone networks for feature extraction and directly perform classification and bounding box regression on the extracted feature maps. In contrast, two-stage detectors, such as R-CNN [16], Fast R-CNN [17], and the popular Faster R-CNN [5], upon which many recent FSOD methods are built, usually employ ResNet [18] as the backbone. In recent years, Transformer [19] technology, which has achieved significant success in natural language processing, has also been successfully applied to object detection. Notably, DETR [20] has been introduced. However, these detectors typically struggle to achieve satisfactory results in Few-Shot scenarios due to the need to handle annotations for a large number of object instances in practical applications.

2.2 Few-Shot Object Detection

FSOD is a significant issue in the field of computer vision. There are several mainstream approaches to address this problem: 1. Metric learning 2. Fine-tuning 3. Meta-learning and 4. Transfer learning. Transfer learning refers to the pre-training of network weights on a baseline dataset to enhance generalization ability in new domains with limited data. When applied to specific tasks, only a few iterations are needed to achieve excellent performance in new tasks. Taking the work of Yan et al. [3] as an example, they adopted a meta-learning framework for FSOD, using Faster/Mask R-CNN as the base and proposing Meta R-CNN. This method includes a support branch for acquiring category attention vectors, which, combined with RoI features, extracts new predictive features for object detection. In other approaches, TFA [21] considers the Faster R-CNN backbone network as category-agnostic, requiring only fine-tuning of the detector's last layer (including category classification and bounding box regression) to transfer feature information from base classes to new classes and achieve performance far beyond previous methods. To address misclassification issues, FSCE [22] introduces contrastive learning based on TFA to optimize the feature embedding space, making instances of the same category closer in feature space and instances of different categories farther apart. In previous studies, researchers used GANs [5] with dual-channel convolution, discriminator, and generator to generate additional information for data augmentation. However, GANs lack effective capabilities for extracting hidden features. Therefore, we chose to use VAE for feature extraction on the support branch's data images.

3 Method

In this section, we first introduce the problem setting regarding FSOD and then describe our motivation and the main architecture of the network.

3.1 Problem definition

This paper uses the FSOD code collection MMfewshot[23] and follows the standard settings of FSOD based on meta-learning in this work. Assuming there is a training dataset $D_{base} \cup D_{novel}$, the data is divided into two classes, one is C_{base}, and the other is D_{base}. The base class and the novel class are two disjoint categories. The training D_{base} contains sufficient C_{base}. FSOD aims to detect objects from both the base dataset D_{base} (containing a large number of annotated objects from C_{base}) and the new dataset D_{novel} (containing very few annotated objects from C_{novel}) by learning. In the K-shot object detection task, each class from C_{novel} has K annotated object instances, but the number is relatively small compared to $K \cdot| C_{novel}|$. For meta-learning, M_{init} is first trained on D_{base} to obtain a basic model, denoted as M_{base}. Typically, a scenario training scheme is used, where each e simulates an N-way-K-shot setting, called meta-training. In each e (also known as a few-shot task), the model is trained on a random subset $D^{e}_{meta} \cup D_{base}$ of K samples $| D^{e}_{meta} |= K \cdot N$ from N classes. Therefore, the model typically needs to learn how to classify based on the input. Finally, the model M_{base} is fine-tuned through meta-fine-tuning for the final task training, obtaining M_{final}. In this project, the finally trained model is applied to a test dataset containing both novel and base classes. The process is shown in Formula 1.

$$\boldsymbol{M}_{init} \xrightarrow[\mathrm{e}=1\cdots E]{D_{\mathrm{meta}} \subset \mathrm{D}_{\mathrm{base}}} \boldsymbol{M}_{base} \xrightarrow{D_{funetune}} \boldsymbol{M}_{final} \tag{1}$$

3.2 Motivation

We use two-stage Faster-RCNN network as our backbone network. Based on previous research findings, this network's performance is not satisfactory for datasets with significant scale variations like the COCO dataset, objects often vary in size, leading to many false positives and negatives. As objects in Pascal VOC images exhibit less variation in size, and thus using only the C4 layer of the backbone network proves effective. To better detect small objects in FSOD, we need to add a network to better extract images of different scales. The FPN upsamples feature maps from each stage in a top-down path to generate multi-scale feature maps. Bottom-up paths enhance the features at each stage by connecting them horizontally, thus helping network learning to make accurate location and category predictions about the target. Therefore this paper introduces the FPN to be used in conjunction with existing meta-learning methods. Meanwhile, we found that FSOD algorithms work on mining the detail information of support branches and query branches and utilize it for transferring into the detection header. We created a mixed feature processing module, in FSOD, hidden features must be extracted from the dataset to make the information transmitted into the detection head richer, in order to further mine the information in the support set, we merge the VAE model into the support branch of the meta-learning model to enable the network to learn more features from the support set. By learning distributions in the latent space, VAE helps to capture shared and differential features between categories, making the model better adapted to new categories. This capability allows the model to perform well in the face of both limited samples and new categories, providing strong support for FSOD tasks.

3.3 Proposed Method

In this paper, the focus is on meta-learning-based methods. However, on the COCO dataset, traditional meta-learning-based methods cannot achieve comparable performance to transfer learning-based methods, and this difference can be attributed to the fact that meta-learning-based methods utilize only the C4 layer for RoI pooling. To bridge this gap, this paper adds FPN to the support branch of meta-learning-based methods since transfer learning-based methods use a FPN to enhance multi-scale feature extraction. This network is constructed on the basis of Meta-RCNN, and its meta-network actually shares the same backbone as Faster-RCNN, which contains two key

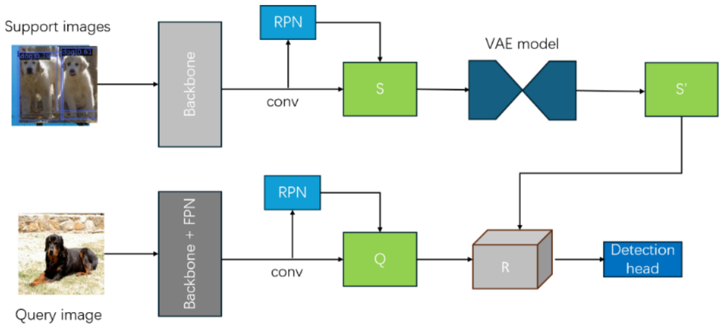

Fig. 1. Mixed feature processing model

branches, the query branch and the support branch. Due to the introduction of the FPN, we enhance the experimental results on the VOC dataset. However, in this paper, we choose to add the FPN to the query branch only, due to the complex feature fusion problem involved in support branches to generate multiple features. With the introduction of this innovation, the model performance in this paper is significantly improved. In addition, emphasize the aggregation between support features and query features. We built a powerful feature extraction module. Figure 1 shows the general structure of the network in this work.

This network, based on Meta-RCNN as the baseline, divides the network into two parts: the support branch and the query branch. Except for adding the FPN network to the query branch, the backbone network is shared between the two branches. This work employs the ResNet101 network to construct the backbone network, as shown in Figure 1. The input of the support set is annotated images, while the input of the query set is original images without any annotations. The VAE model is introduced into the support branch to extract deeper semantic information from the support set. Initially, the images of the support set are input into the backbone, and the features S of the support set are obtained through RPN. Then, the features S are input into the VAE module, where they are transformed into a class distribution N using the encoder of the VAE module. Here, N follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$, and variational features z are sampled from N. Finally, the features S' are output through the decoder. The obtained S' and the query features Q from the query branch are input into the R module for the final feature fusion step. The fused results are then input into the detection head for subsequent detection. This paper tests several types of R modules and identifies the optimal approach, which improves the performance effectively.

VAE Module. Previous work often encoded support examples as single feature vectors that were difficult to represent the entire class distribution. Especially when data is scarce and instances vary widely, it is challenging to accurately estimate the class center. Previous research has also introduced GANs into object detection modules and achieved good results. GANs consist of two branches: a generator network and a discriminator network. The generated images are very similar to the original images but lack exploration and analysis of the latent space. Therefore, this chapter starts with the VAE model, which can explore the latent space more effectively than GAN networks. Inspired by recent advances in variational feature learning, we transform support features into outputs with VAE. Then, we use samples of the output features for robust feature aggregation.

During the encoding phase, VAE receives input from images in the support set and maps the input data to the latent space through an encoder. The encoder, consisting of neural networks, maps the input data to the mean and variance parameters in the latent space. These two parameters define a latent distribution, typically assumed to be a multivariate normal distribution. Specifically, given input data x, the encoder produces the mean μ and variance σ of the latent variable z. This latent variable z is a point sampled from the latent distribution, representing the representation of the input data in the latent space. The encoder's task is to learn a mapping function that maps the input data to the parameters of the latent distribution. During the decoding phase, VAE uses the latent variable z o generate reconstructed data x' through a decoder. The decoder, also a neural network, takes the latent variable z and attempts to restore the input data. Since this chapter introduces randomness into the latent distribution during the encoding phase, in the decoding phase, this paper samples a point from the latent distribution to generate multiple possible reconstruction results. Therefore, the VAE model also serves to diversify the data and alleviate issues such as the scarcity of samples and the resulting low model accuracy in FSOD.

Specifically, the operation of the VAE is as follows: Given a real sample $\mathrm{X_k}$, suppose there exists a distribution $\mathrm{p(Z|X_k)}$ (posterior distribution) specific to $\mathrm{X_k}$. It is typically assumed to be a normal distribution, and $p(Z \mid X)$ is aligned with a standard normal distribution to prevent noise from being zero while ensuring that the model has generative capability.

In this process, the mean and variance of the distribution are constructed through two neural networks, $\mu_k = f_1(X_k)$ and $\log \sigma_k = f_2(X_k)$,and the latent variable Z is sampled from it. This dedicated distribution is a hypothesis of the posterior distribution, used to train a generator $X = g(Z)$, where the generator's task is to sample a Z_k from the distribution $p(Z|X_k)$ and then generate $\widehat{X_k} = g(Z_k)$ through a generator. The entire training process is completed by minimizing the reconstruction error $D\left(\widehat{X_k}, X_k\right)$ to ensure that the generator can effectively reconstruct the input samples. Through this process, VAE achieves the mapping from the latent space to the data space, making points in the latent space correspond to reasonable representations of the data. The structure of this latent space is continuously optimized during the training process of VAE to better capture the features of the data.

$$L_{\mu,\sigma^2} = L_\mu + L_{\sigma^2} \tag{2}$$

$$L_\mu = \frac{1}{2}\sum_{i=1}^{d} \mu_{(i)}^2 = \frac{1}{2}|f_1(X)|^2 \tag{3}$$

$$L_{\sigma^2} = \frac{1}{2}\sum_{i=1}^{d}\left(\sigma_{(i)}^2 - \log \sigma_{(i)}^2 - 1\right) \tag{4}$$

$$L_{\mu,\sigma^2} = \frac{1}{2}\sum_{i=1}^{d}\left(\mu_{(i)}^2 + \sigma_{(i)}^2 - \log \sigma_{(i)}^2 - 1\right) \tag{5}$$

The KL divergence between the normal distribution and the standard normal distribution,$\mathrm{KL}\left(\mathrm{N}(\mu, \sigma^2)|\mathrm{N}(0,1)\right)$, serves as this additional loss, ensuring that all $\mathrm{P(Z|X)}$ align with the standard normal distribution by computing the KL loss function. Constraining the latent distribution to a standard normal distribution helps to make the structure of the latent space more regular and uniform, improving the model's generalization ability.

$$p(Z) = \sum_X p(Z \mid X)p(X) = \sum_X N(0,1)p(X) = N(0,1)\sum_X p(X) = N(0,1) \tag{6}$$

It ensures that the learned representations are more consistent across the entire latent space and contributes to more stable model training. It can be seen as a form of regularization that helps prevent overfitting while making the model easier to optimize.

R Module. When receiving the outputs S' from the VAE model and Q from the backbone network, this chapter explored various feature fusion strategies. Different fusion strategies produce different outcomes. Drawing from prior research, contrastive learning and metric learning were introduced, employing cosine similarity to calculate intra-class and inter-class features. This elevated the model's complexity. The chapter examined several aggregation strategies to consolidate the output features.

Scheme 1 and Scheme 2 only utilize fully connected layers that play a role in the fine-tuning method. Firstly, the outputs S' from the VAE and Q from the query branch are separately fed into two fully connected layers. Subsequently, the outputs of these two fully connected layers are directly summed up. They are aggregated together using the following formula, and then the aggregated F is inputted into the detection head for detection.

$$F_{S'} \oplus F_Q = F_1 \tag{7}$$

The second scheme is similar to the first one, where the outputs S' from the VAE and Q from the query branch are separately fed into two fully connected layers. Subsequently, the outputs of these two fully connected layers are directly multiplied (dot product). They are aggregated together using the following formula, and then the aggregated F is inputted into the detection head for detection.

$$F_{S'} \otimes F_Q = F_2 \tag{8}$$

The third scheme, as illustrated in Figure 2, involves feeding the outputs S' from the VAE and Q from the query branch into two backbone networks for further feature extraction. Each branch consists of a sequence of convolutional layers with kernel sizes of 5×5, 3×3, and 1×1, respectively. These layers all utilize Leaky ReLU as the activation function. The convolutional layers with kernel sizes of 5×5 and 3×3 are employed for feature extraction. In the final stage, concatenation operations (Concat) are used to interchange and merge the features of S' and Q After the last concatenation operation, the fused image is obtained through a convolutional layer with a 1×1 kernel size, activated by the Sigmoid function.

In order to ensure that the image size remains unchanged during the generation process, the stride for all convolution operations is set to "1", while the padding value is set to half of the kernel size in the corresponding convolution layer.

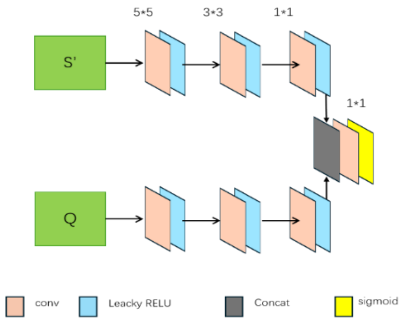

Fig. 2. R network architecture diagram 3

In Table 1, the three values in parentheses corresponding to Conv represent the number of input channels, the number of output channels, and the kernel size of the convolution, respectively. Batch normalization (BN) denotes batch normalization of the data.

Table 1. R Module Network Architecture 1

Network Layers	Convolutional Parameters	Activation Function	Network Layers	Convolutional Parameters	Activation Function
X1	(1,16,5)	LeakyRelu	Y1	(1,16,5)	LeakyRelu
X2	(16,32,3)	LeakyRelu	Y2	(16,32,3)	LeakyRelu
X3	(32,16,1)	LeakyRelu	Y3	(32,16,1)	LeakyRelu
		Concat			
X4	(32,1,1)	Sigmoid			

Loss Function. In the VAE, this chapter only introduced one encoder and one decoder. The encoder consists of a linear layer and two parallel linear layers, which produce μ and σ respectively. The decoder consists of two linear layers, generating the reconstructed features S'. All layers in this chapter maintain the same dimensions, adopting end-to-end training, and incorporating the following multi-task loss:

$$L = L_{rpn} + L_{\text{meta}} + L_{\text{cons}} + \alpha L_{KL} \tag{9}$$

Here, L_{rpn} represents the total loss of the Region Proposal Network (RPN): L_{rpn}
.

$$L_{rpn}(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{rgg}(t_i, t_i^*) \tag{10}$$

L_{reg} is the regression loss:

$$L_{reg}(t_i, t_{i^*}) = \sum_{j \in \{x,y,w,h\}} Smooth_{L1}\left(t_j^i - t_j^{i'}\right) \tag{11}$$

α represents the weight coefficient (default $\alpha = 2.5 \times 10^{-4}$). This chapter directly minimizes the KL divergence $L_{\mathrm{KL}}(p(x)|q(x))$, where λ is the weight of the two parts of the loss, N_{cls} and N_{reg} are the numbers of foreground and background samples, and L_{meta} is the meta-loss. Given a RoI feature Z_{ij} , to avoid prediction ambiguity after soft attention, different attention vectors for objects of different classes will have different feature selection effects on Z_{ij}. Therefore, a simple meta-loss L_{meta} is used to disperse the object attention vectors inferred in meta-learning. It is implemented through cross-entropy loss, aiming to encourage diversity in class features, which enhances the performance of the network in this paper. Our work applies consistency loss L_{cons} to the reconstructed feature S', which is defined as the cross-entropy between S' and its class label c:

$$L_{\mathrm{cn}} = L_{CE}\left(F_{cls}^{S'}\left(S'\right), c\right) \tag{12}$$

4 Experiments

4.1 Dataset and Experimental Setting

We utilized the PASCAL VOC dataset[24] to evaluate our method. K-shot object detection was performed on three partitions of novel/base classes, where K = (1, 2, 3, 5, 10). In the PASCAL VOC dataset, we organized the 20 classes into three splits, each containing 15 base classes and 5 novel classes. For each novel class set, we conducted experiments with K={1, 2, 3, 5, 10} shot settings. The model was trained on the base classes and tested on the novel classes for each split.

4.2 Few-Shot Object Detection Results

Dilution experiment of the FPN module. Similarly, this chapter conducted ablation experiments on whether to include FPNs using data from Pascal VOC split 1. It was found that the improvement on the Pascal dataset was more pronounced. Through this ablation experiment, we found that the recognition accuracy of new categories was improved by about 2-3 % with FPN compared with that without FPN.

Differential ablation experiments of various R modules. This chapter employed three different aggregation methods to validate the effectiveness of the experiments and found that different processing methods lead to different outcomes.

Table 2. Ablation experiments on FPN conducted on VOC split1

VOC split1		
FPN	3	5
×	60.1	63.8
√	61.3	65.3

Contrastive learning and metric learning were introduced in previous studies mainly to calculate the relevance and irrelevance between different classes. In our work, we aimed to maximize the utilization of the VAE outputs. Therefore, we directly fed the results from the query branch and support branch into fully connected layers for feature aggregation, using two aggregation methods (Scheme 1 as Equation 7 aggregation method, Scheme 2 as Equation 8 aggregation method). Considering the dual-branch structure of the previous network, a third experiment was conducted. In Scheme 3, the results from the support branch and query branch were separately sent to two branches through a dual-channel network. After passing through the convolutional network, feature aggregation was performed. Finally, it was found that Scheme 1 yielded the best results.

Table 3. Dilution experiment of different R modules

Method	Novel Set1					Novel Set2					Novel Set3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
1	51.2	61.0	61.3	63.7	65.3	34.3	43.1	47.1	50.2	52.3	47.4	45.2	47.7	51.4	55.3
2	48.7	58.0	58.7	61.7	63.8	32.5	40.6	42.9	47.3	50.9	31.0	43.9	45.6	48.8	51.0
3	49.2	59.8	60.2	62.3	63.9	32.6	41.6	43.8	47.9	51.0	38.6	43.9	46.1	49.8	53.5

Few-Shot Object Detection Results and Visualization. For each novel class set, experiments are conducted with settings of K=1, 2, 3, 5, and 10 shots. The model is trained on the base classes and tested on the novel classes in each partition. From Table 4, we can see that the accuracy of our model is higher than most works, and has excellent performance, our results are the best in Novel set1 and 2, lacking in Novel set3, but still good.

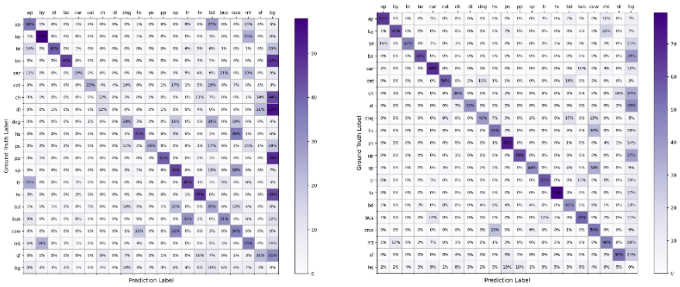

Fig. 3. Confusion matrix of the mixed feature processing model: (a):VOC split1 1shot (b): VOC split1 10 shot

We selected split1 with 1-shot and 10-shot settings to generate confusion matrices for visualizing the results Fig 5. The corresponding abbreviations for these categories are as follows: ap-aeroplane, by-bicycle, bt-boat, bo-bottle, ch-chair, di-diningtable, hshorse, ps-person, pp-pottedplant, sp-sheep, tr-train, tv-tvmonitor, bd-bird, mt-motorbike, sf-sofa, bg-background. Each row of the matrix represents the actual category, while each column represents the predicted category.

When the row and the column are the same, the detection is correct, the rest is the error detection (positioning or classification error), and the bg (background) intersection is the missed detection situation. Upon comparison, we observed numerous instances of false positives and missed detections in the 1-shot phase. However, these cases were significantly reduced in the 10-shot phase. In the 10-shot phase,

we identified both error detections and missed detections, but the overall detection performance of our model remained superior.

Table 4. The detection results on Pascal VOC by average score

Method/Shots	Novel Set1					Novel Set2					Novel Set3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
MetaDet [25]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
Meta R-CNN[3]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
FSIW[26]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
RepMet[27]	26.1	32.9	34.4	38.6	41.3	17.2	22.1	23.4	28.3	35.8	27.5	31.1	31.5	34.4	37.2
TFA/cos [21]	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
TFA/fc[21]	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
FSCE [22]	32.9	44.0	46.8	52.9	59.7	23.7	30.6	38.4	43.0	48.5	22.6	33.4	39.5	47.3	54.0
DCNet [28]	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
Retentive[29]	42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1
QA[30]	42.4	51.9	55.7	62.6	63.4	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5
FADI[31]	50.3	54.8	54.2	59.3	63.2	30.6	35	40.3	42.8	48	45.7	49.7	49.1	55	59.6
Ours	51.2	61.0	61.3	63.7	65.3	34.3	43.1	47.1	50.2	52.3	47.4	45.2	47.7	51.4	55.3

5 Conclusion

In our work, the Meta R-CNN network is divided into two branches: the support branch and the query branch. We named this model as mix feature processing model, FPN is introduced in the query branch to enhance the ability of the network to extract features of different dimensions. In the support branch, VAE is introduced for feature extraction to realize data expansion. We also introduce *R* module to do feature aggregation of the output supporting branches and query branches, ablation experiments are done to verify the effectiveness of our feature aggregation method and add it to the model, use VOC data set to verify our network. In comparison and analysis with other FSOD methods, our method stands out and gets good results.

Acknowledgments. This study is supported by the National Natural Science Foundation of China (Nos. 62101481 and 62261060), Yunnan Fundamental Research Projects (Nos. 202201AT070112, 202301AW070007, 202301AU070210, 202401AT070470, and

202301AT070407), Major Scientific and Technological Project of Yunnan Province (No. 202202AD080002), and National Double Tops Construction Fund of China (C619300A020).

References

1. Jin X, Hou J, Zhou W, et al.: Recent advances in image fusion and quality improvement for cyber-physical systems. Frontiers in Neurorobotics **17**, 1201266 (2023)
2. Jin X, Wang R, Lee S J, et al.: Adversarial attacks on multi-focus image fusion models. Computers & Security **134**, 103455 (2023)
3. Yan X, Chen Z, Xu A, et al.:Meta r-cnn: Towards general solver for instance-level low-shot.In:Proceedings of the IEEE/CVF International Conference on Computer Vision,pp.9577-9586(2019)
4. Ren S, He K, Girshick R, et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems **28**, (2015)
5. H. Lee, S. Kang and K. Chung.:Object Detection with Dataset Augmentation for Fire Images Based on GAN.In: 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022,pp. 2118-2123(2022)
6. Noh J, Bae W, Lee W, et al.:Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection.In:Proceedings of the IEEE/CVF International Conference on Computer Vision,pp. 9725-9734(2019)
7. Bai Y, Zhang Y, Ding M, et al.:Sod-mtgan: Small object detection via multi-task generative adversarial network.In:Proceedings of the European conference on computer vision (ECCV),pp. 206-221(2018)
8. Li J, Liang X, Wei Y, et al.:Perceptual generative adversarial networks for small object detection.In:Proceedings of the IEEE conference on computer vision and pattern recognition,pp.1222-1230(2017)
9. Kingma D P, Welling M.: Auto-encoding variational bayes. arXiv preprint arXiv **1312.6114**, (2013)
10. Liu W, Anguelov D, Erhan D, et al.:Ssd: Single shot multibox detector.In:Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings,pp. 21-37.(2016)
11. Redmon J, Farhadi A.: YOLO9000: better, faster, stronger.In:Proceedings of the IEEE conference on computer vision and pattern recognition,pp. 7263-7271(2017)
12. Redmon J, Farhadi A.: Yolov3: An incremental improvement. arXiv preprint arXiv **1804.02767**, (2018)
13. Bochkovskiy A, Wang C Y, Liao H Y M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv **2004.10934**, (2020)
14. Girshick R, Donahue J, Darrell T, et al.: Rich feature hierarchies for accurate object detection and semantic segmentation.In:Proceedings of the IEEE conference on computer vision and pattern recognition,pp.580-587(2014)
15. Girshick R.: Fast r-cnn.In:Proceedings of the IEEE international conference on computer vision,pp.1440-1448(2015)
16. Girshick R, Donahue J, Darrell T, et al.: Rich feature hierarchies for accurate object detection and semantic segmentation.In:Proceedings of the IEEE conference on computer vision and pattern recognition,pp.580-587(2014)

17. Girshick R, Donahue J, Darrell T, et al.: Rich feature hierarchies for accurate object detection and semantic segmentation.In:Proceedings of the IEEE conference on computer vision and pattern recognition,pp.580-587(2014)
18. He K, Zhang X, Ren S, et al.: Deep residual learning for image recognition.In:Proceedings of the IEEE conference on computer vision and pattern recognition,pp.770-778(2016)
19. Vaswani A, Shazeer N, Parmar N, et al.: Attention is all you need. Advances in neural information processing systems **30**, (2017)
20. Carion N, Massa F, Synnaeve G, et al.: End-to-end object detection with transformers.In:/European conference on computer vision,pp.770-778.Springer International Publishing(2016).
21. Wang X, Huang T E, Darrell T, et al.: Frustratingly simple FSOD. arXiv preprint arXiv **2003.06957**, (2020)
22. Sun B, Li B, Cai S, et al.: Fsce: Few-shot object detection via contrastive proposal encoding.In:Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,pp. 7352-7362(2021)
23. Openmmlab few shot learning toolbox and benchmark,url **https://github.com/open-mmlab/mmfewshot**.Last accessed 2021
24. Everingham M, Van Gool L, Williams C K I, et al.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**, 303-338(2010)
25. Wang Y X, Ramanan D, Hebert M.: Meta-learning to detect rare objects.In:Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,pp. 9925-9934(2019)
26. Xiao Y, Lepetit V, Marlet R.:Few-shot object detection and viewpoint estimation for objects in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 3090-3106(2022)
27. Karlinsky L, Shtok J, Harary S, et al.:RepMet: Representative-based metric learning for classification and one-shot object detection. arXiv preprint arXiv **1806.04728**, (2018)
28. Hu H, Bai S, Li A, et al.:Dense relation distillation with context-aware aggregation for few-shot object detection.In:Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,pp. 10185-10194(2021)
29. Fan Z, Ma Y, Li Z, et al.: Generalized few-shot object detection without forgetting.In:Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,(2021)
30. Han G, He Y, Huang S, et al.: Query adaptive few-shot object detection with heterogeneous graph convolutional networks.In:Proceedings of the IEEE/CVF International Conference on Computer Vision,pp.3263-3272(2021)
31. Cao Y, Wang J, Jin Y, et al.:Few-shot object detection via association and discrimination. Advances in neural information processing systems **34**,16570-16581 (2021)