International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 7, Number 3, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v7n3.04
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Calibrating Fake Feature Statistical Distribution for Few-Shot Object Detection

Lanlan Liu [1], Haifeng Sima [2, *]

[1] Faculty of Arts and Law, Henan Polytechnic University, Jiaozuo, China
[2] School of Software, Henan Polytechnic University, Jiaozuo, China
*Corresponding Author: smhf@hpu.edu.cn

ABSTRACT

The booming development of object detection has benefited from rich datasets. However, obtaining a large amount of labeled data for scarce objects is expensive. Few-Shot Object Detection (FSOD) has garnered significant attention, aiming to learn new categories with only a small number of labeled samples. In this paper, we propose a novel FSOD model based on data augmentation, namely Calibrating Fake Feature Statistical Distribution for Few-Shot Object Detection (CFSD). First, we utilize feature-specific generative adversarial methods to synthesize fake features, addressing the issue of overfitting in the detection head and classifier when dealing with limited samples. However, due to the model's insufficient training samples, the distribution of the generated fake features in the feature space tends to lean towards the base category. To tackle this issue, we design a distribution calibration module to adjust the statistical distribution of fake features, aligning it with that of novel category features. Finally, we conducted tests on the Pascal VOC and MS COCO datasets, and the results unequivocally confirmed the effectiveness of our method. The test results outperform existing FSOD-based methods and other data augmentation-based methods.

KEYWORDS

Meta-learning; Data augmentation; Distribution calibration; Generative adversarial network

1. INTRODUCTION

Traditional object detection methods usually require a large amount of labeled data to ensure that the model can accurately identify and locate the objects in the image. However, in practical applications, acquiring large-scale and high-quality labeled data is often a complex and expensive task for researchers. Specifically, the availability of labeled data in emerging fields or specific tasks is extremely limited. Therefore, this has spurred the study of few-shot object detection. The primary task of few-shot object detection [1-5] is to address the challenges posed by insufficient labeled data.

Data augmentation [6-8] is one method to address the few-shot problem. It not only expands the number of samples in each category to increase sample diversity but also enables the creation of new categories and tasks. Among these methods, generative models are considered efficient for data augmentation. This method simulates the distribution of real samples to generate diverse fake samples. Currently, Variational Auto-Encoders (VAE) [9-11] and Generative Adversarial Networks (GANs) [12-15] are commonly used generation technologies. VAE consist of an encoder and a decoder. The encoder transforms the input data into a lower-dimensional latent variable, serving as an abstract representation of the data. This latent variable captures key features and the object distribution within the data, allowing the model to better comprehend the original data. The decoder then maps these latent variables back to the original data space to generate new samples. The main idea of GANs

involves the confrontation between the generator and the discriminator in a game process. The generator aims to produce realistic data to deceive the discriminator, which distinguishes between real and fake data. However, it is challenging for generators to learn from a limited number of samples. GANs typically use implicit feature generation, where the generation process is contained in the trained generator. With sufficient data, the generator can naturally better reveal the true distribution of the data. As training samples decrease, it becomes increasingly difficult for the generator to model the statistical distribution of new class features. This, in turn, makes it challenging for the model to accurately determine the decision boundary of the classifier. Hence, it is essential to rectify samples with indistinct classification boundaries, and assign them to a distinct cluster center. Distribution calibration corrects the statistical distribution of features to ensure alignment with the statistical distribution of cluster centers when handling samples with ambiguous decision boundaries. By this means, the model becomes adept at capturing the intrinsic patterns within the data, thereby enhancing its capability to process samples with unclear decision boundaries. However, distribution calibration necessitates computing the similarity between each new class and all base classes. In scenarios with a substantial number of visible classes, distribution calibration exhibits high space dimension and computational complexity at runtime. Moreover, there might not always be a base class resembling the target new class within the visible classes. If the sample class encompasses a broad span, the achieved effect may be less satisfactory.

This paper introduces a novel data augmentation approach, namely CFSD. It is implemented based on a meta-learning framework, and mainly comprising a generative adversarial module and a distribution calibration module. These modules aim to address two main challenges: (1) Ineffective generalization of the model's classifier and regressor due to limited data. (2) Deviation of the generated fake features from the real cluster center. Specifically, the generative adversarial module is used to generate fake features. Different from EBGan [16], the generative adversarial module does not generate feature samples with the same dimensions as the original samples, but generates high-level fake feature vectors. This means that the generator and discriminator discard a large number of convolution operations and upsamps, simplifying the complexity of feature generation and allowing the model to extract a richer representation of features. However, in the training of new categories with a limited number of examples, the EBGan model introduces a challenge where the generated features significantly deviate from the statistical distribution of object features. This deviation can hinder the model's ability to accurately determine the decision boundary of category samples. To address this issue, the distribution calibration module [30-32] is employed to realign the statistical distribution of fake features, ensuring consistency with the distribution of support features. This enhances the compatibility of fake features with target tasks. Differently, our mothed doesn't transfer the distribution of the base category to the new task, instead, it focuses on the statistical distribution of support features. This ensures that the fake features align with the distribution of support features, and reduce the extensive calculation of similarity. Additionally, our distribution calibration method is implemented on a pre-trained feature extractor, which eliminating the need for fine-tuning or retraining the feature extractor. The contributions of this paper can be summarized as following three points:

(1) This paper introduces a few-shot object detection model within a meta-learning framework, which focusing on feature generation. The objective is to generate synthetic features based on a small number of labeled samples, and address the challenge of overfitting in detection heads and classifiers caused by limited samples.

(2) The paper proposes a distribution calibration module to realign the distribution of fake features in the feature space, ensuring that the mean and covariance of fake features align with those of support features. This module resolves the difficulty the model faces in determining the decision boundary of category samples.

(3) Through comprehensive ablation and comparison experiments, we have scientifically validated the effectiveness of our method. The model demonstrates superior detection performance on both the

Pascal VOC dataset and the MS COCO dataset compared to the general FSOD model. The average test results across each dataset show an improvement of 1% to 2% compared to the baseline.

2. RELATED WORK

2.1. Few-shot Object Detection

Currently, few-shot object detection frameworks can be categorized into two main groups: transfer learning and meta-learning. Transfer learning [17-20] aims to apply knowledge acquired from a source task to a new task. This knowledge is typically gained through pre-trained neural network models, with pre-trained weights containing common feature representations. During training for the target task, the model fine-tunes the classifier and regressor to adapt to new categories. However, the effectiveness of transfer learning may be limited if there is significant domain dissimilarity between the source and target tasks, a challenge that transfer learning continues to address. On the other hand, Meta-Learning [21-25] represents a more advanced approach. Its fundamental concept involves the model simulating multiple diverse tasks during the training process, enabling it to quickly adapt to various tasks. Unlike transfer learning, meta-learning doesn't require the model to achieve high performance in a specific task but focuses on capturing meta-knowledge across all tasks. This meta-knowledge, particularly valuable when dealing with a small number of available samples, allows the model to "learning to learn" Consequently, in cross-domain detection problems, meta-learning tends to outperform transfer learning. In summary, both methods aim to tackle few-shot tasks by leveraging limited knowledge and reducing dependence on new data, ultimately enhancing the performance of target tasks. In this paper, our model is implemented within a meta-learning framework.

2.2. Data Augmentation-Based Few-Shot Approaches

The essential issue of few-shot tasks is the scarcity of samples. Initially, researchers used data augmentation to solve this problem. Liu et al. [26] increased the number of samples by rotating the original images by 90 degrees, 180 degrees, and 270 degrees, allowing the model to sample more task instances during training. However, the effect of data preprocessing methods is limited, and the expanded samples are still limited by the diversity of the original data. The researchers then expand the sample by generating models, which can be generated in a variety of ways, including diffusion models, generative adversarial networks, and VAE. Diffusion models have been shown to synthesize higher quality fake samples. Trabucco et al [27] used a text-to-image diffusion model (DA-Fusion) to generate diverse fake images. This method adapts the diffusion model to new domains by inserting and fine-tuning new tokens in text encoders representing visual concepts. In addition, Cheng et al. [28] adopted CenterNet as the basic framework and redesigned the features to enable the model to overcome knowledge forgetting to a large extent and adapt to unseen knowledge. However, original diffusion models are slow to sample, often requiring thousands of evaluation steps to draw a single sample. Therefore, it is difficult for diffusion models to generalize to various scenarios. The generative adversarial network synthesizes fake data through the game of generator and discriminator. Li et al [15] used cWGAN to synthesize fake datasets to increase the number of training samples. At the same time, classification regularization and anti-collapse regularization are introduced to promote the discriminability and diversity of synthetic features. But the success of generating adversarial networks often depends on having a large number of training samples. If there are too few samples, the generator cannot capture the distribution of the new class samples in the feature space, and the generated fake sample will deviate from the real clustering center, resulting in the model cannot divide the effective decision boundary. VAE belongs to the same generative model as GAN. Differently, VAE combines the ideas of autoencoders and probabilistic graph models, aiming to learn the potential representation of data and the generative distribution of data. Han et al [29] use VFA variational autoencoder to estimate the distribution of categories and sample variational features from the variance distribution of base class samples. The VFA model applies VAE to the few-shot task for the

first time. However, the reconstruction loss of VAE often results in the generated samples being too smooth, and thus the quality of the generated fake samples is usually low. In this paper, we calibrate the mean and covariance of the generated features based on the GAN model to produce high-quality fake features.

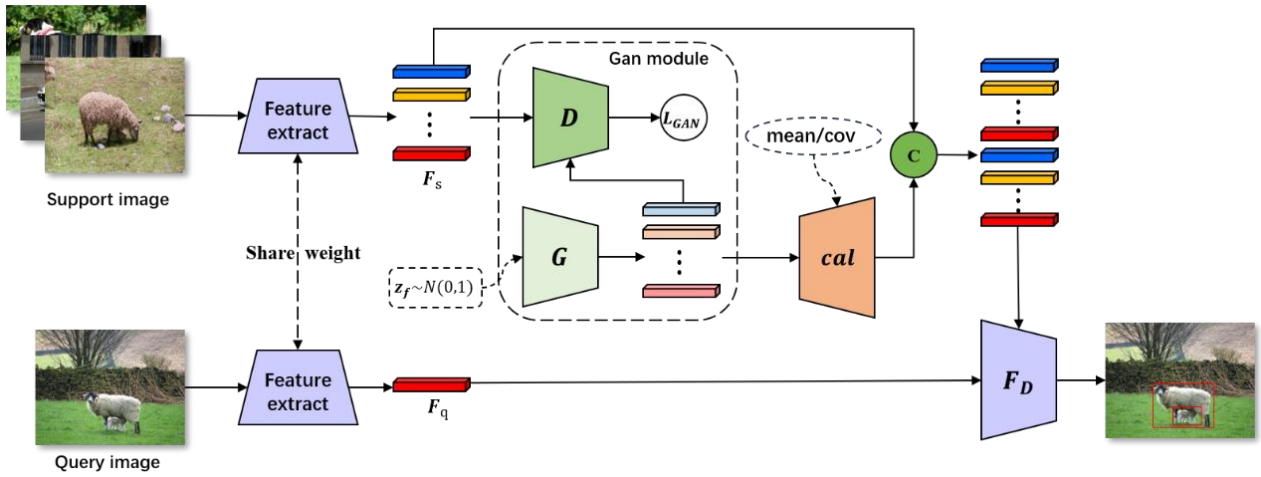

Figure 1. Overall structure of CFSD model

2.3. Data Augmentation-Based Few-Shot Approaches

The distribution of features in the feature space significantly influences the model's decision-making effectiveness. Fuzzy boundaries between categories can pose challenges for the model in classification tasks. Therefore, calibrating fuzzy category samples can clarify the boundaries between categories, which is beneficial to model classification. Zhang et al [30] introduced the Task Coding with Distribution Calibration (TEDC) model, which reduces intra-class distribution differences by minimizing the distance between support features and query features within the same category, thereby improving the model's performance. Park et al [31] attempted to enhance feature representation by exploiting intra-class variance. Yang et al [32] chose the base category with the highest similarity to the target feature, and directly calibrate the distribution estimate of Gaussian noise based on the statistical information of this category. In our method, we diverge from using the statistical information of base categories and instead leverage the distribution of novel categories as the foundation. This enables us to generate more reliable feature samples by calibrating the statistical distribution of fake features.

3. METHOD

First, we introduce the training process of the meta-learning framework in Section 3.1. Secondly, Section 3.2 shows the overall architecture of this model. Finally, the structure and implementation of the generative adversarial module and distribution calibration module are discussed in detail in Sections 3.3 and 3.4.

3.1. Problem Definition

This paper divides the categories of the dataset into two parts: the base classes C_{base} with abundant annotated samples and the novel classes C_{novel} with only a few annotated samples. C_{base} and C_{novel} do not contain the same class, i.e., $C_{base} \cap C_{novel} = \emptyset$. The meta-detection model uses the generalizable knowledge in C_{base} to fine-tune C_{novel}, allowing the model to detect objects from both C_{base} and C_{novel}. In the novel class, the number of categories and samples is set to N-way and K-shot, indicating N categories and each category has K samples.

3.2. Model Overview

The CFSD model is implemented based on the meta-learning framework. The high-level feature vectors after deep convolution processing contain rich semantic information and statistical information. As a result, there is no need for extensive additional convolution processing. In this article, the model utilizes a lightweight generative adversarial module (GAN module) to capture the statistical distribution of high-level features, aiming to generate more diverse fake features [44, 45]. In this paper, the generative adversarial module utilizes a lightweight generative network to capture the statistical distribution of features. Second, the Distribution Calibration module is employed to adjust the distribution of fake features in the feature space, and ensure them share the same cluster center as the real features. In this model, the calibrated fake features and original features are used together to train the model's classifier. This approach enhances the diversity of data and aids in improving the generalization performance of the detector when data samples are limited.

Fig. 1 illustrates the specific structure of the CFSD model. Initially, the model employs two feature extractors with the identical structure to extract query features F_q and support features F_s, respectively. These feature extractors utilize the ResNet101 model with shares weights. In the generative adversarial module, the model randomly generates a set of Gaussian noise $z_f \sim N(0,1)$. Subsequently, the generator (G) transforms the Gaussian noise into fake features similar to real features. The discriminator (D) distinguishes whether the input features are real or fake, and the difference between the predicted value and the real value is measured using Mean Squared Error (MSE) and back-propagated. The distribution calibration module (cal) utilizes the mean and covariance of the support features to calibrate the generated fake features, making the distribution of fake features closer to the real features. Finally, the model employs the fake features and support features F_s to jointly train the classifier and regressor of the model.

3.3. Generative Adversarial Module

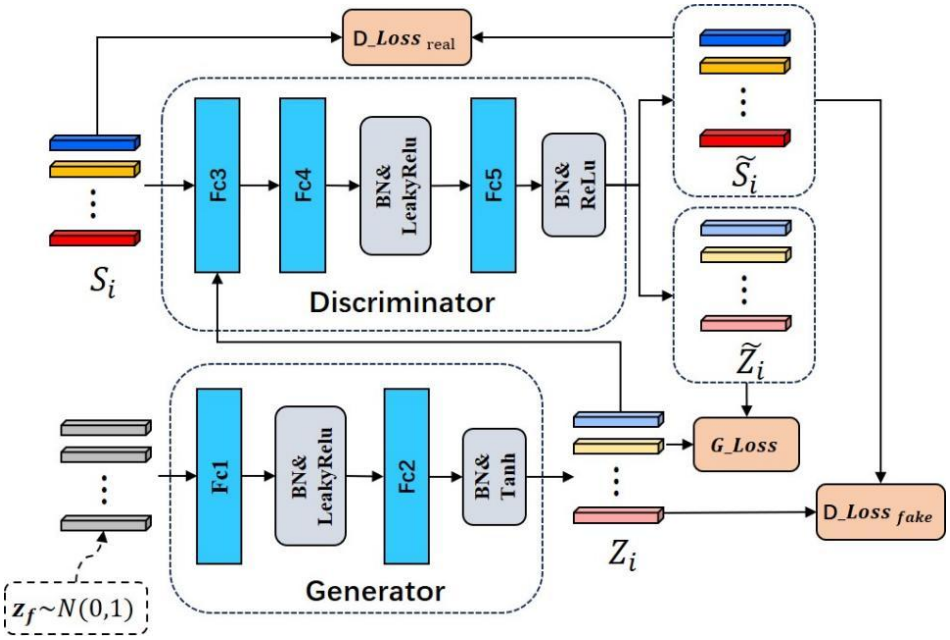

Figure 2. Generative adversarial module structure diagram

The EBGAN model has demonstrated remarkable performance in image synthesis, utilizing autoencoders and numerous convolutions to simulate the original image. However, if the EBGAN model is directly used to generate fake images of the original size for training a few-shot model. It would necessitate extracting features from a shallow network, resulting significant computational resource wastage. At the same time, this approach would inflate the model structure, making it more

challenging to generalize to new categories. Therefore, the synthesis strategy in this paper is a variant of the EBGAN model, which focusing on transforming low-level original images into high-level feature vectors. This adaptation implies the omission of a considerable number of convolution and upsampling operations. The generation module can directly acquire high-level semantic information, reducing the complexity of the module. Additionally, in high-level space, the model can more easily capture the statistical distribution of features, minimizing the impact of irrelevant noise and laying the groundwork for subsequent statistical distribution calibration. Figure 2 offers a detailed overview of the structure of the Generative Adversarial Network module, encompassing a generator for processing Gaussian noise and a discriminator for evaluating feature authenticity. The model's training process involves a dynamic interplay between these two components.

Generator (G). The initialized noise features, $z_f \in R^{C\times 2048}$, follow a standard normal distribution $N(0,1)$. The generator transforms z_f through a non-linear conversion to simulate the distribution of real features. Specifically, the convolutional layers in the generator are replaced by two fully connected layers ($Fc1 \in dim^{1\times 1024}$, $Fc2 \in dim^{1\times 2048}$), enabling the mapping of input features to a high-dimensional feature space. To address the issue of gradient vanishing, the LeakyReLU function is used to activate neurons, with a negative slope parameter set to 0.2. Additionally, the tanh function, combined with Batch Normalization (BN), ensures that the generated data is constrained within the range $(0, 1)$. The entire process can be summarized by Equation 1.

$$Z_i = G(z_f) = tanh\ (BN(Fc2(LeakyReLu(BN(Fc1(z_f)))))),\quad i = 1,2,3,\ldots \tag{1}$$

Discriminator (D). The structure of the discriminator is similar to that of the generator. It takes as input either a real feature ($S_i \in R^{C\times 2048}$) or a fake feature ($Z_i \in R^{C\times 2048}$), and also employs fully connected layers. The primary function is to evaluate whether the input feature is generated or real. Specifically, the discriminator initially utilizes two fully connected layers ($Fc3 \in dim^{1\times 2048}$, $Fc4 \in dim^{1\times 1024}$) to map features into a representation with 1024 neurons. Subsequently, Batch Normalization (BN) combined with the LeakyReLu function facilitates effective gradient updates during backpropagation, mitigating the issue of neurons being activated to 0 while rectifying feature distributions. The final fully connected layer ($Fc5 \in dim^{1\times 2048}$) maps the discriminative features to a dimensionality of 2048. BN cooperates with the ReLu function to normalize the features while making the neurons sparse, maximizing the neuron's filtering capability. Equation 2 shows the specific implementation process of the discriminator:

$$D(X) = ReLu\ (BN(Fc5(LeakyReLu(BN(Fc4(Fc3(X))))))) \tag{2}$$

$$\tilde{Z}_i = D(Z_i),\quad i = 1,2,3,\ldots \tag{3}$$

$$\tilde{S}_i = D(S_i),\quad i = 1,2,3,\ldots \tag{4}$$

$D(\cdot)$ is the discriminator, Z_i and S_i are fake features and real features respectively.

Loss function. In order to mitigate the issue of mode collapse, the loss function Inherited the 'energy' concept from EBGAN [16]. Both the generation loss and the discrimination loss employ Mean Squared Error (MSE) to minimize the difference between fake features and the discriminative results, The loss functions are defined as follows:

$$G\_Loss(Z_i) = MSE(Z_i, D(Z_i)),\quad i = 1,2,3,\ldots \tag{5}$$

$$D\_Loss(S_i, Z_i) = MSE(D(S_i), S_i) + [m - MSE(D(Z_i), Z_i)]^+,\quad i = 1,2,3,\ldots \tag{6}$$

Here, m represents the energy gap between real features and fake features. The generator aims not only to reduce the reconstruction error of real features but also to make the reconstruction error of fake features approach m.

3.4. Distribution Calibration Module

Two sets of similar categories exhibit comparable distributions in the feature space, but the generated features may deviate from the actual distribution range. This module aims to recalibrate the generated fake features to align with the clustering distribution of real data. In this module, instead of calibrating the distribution of the original data, we aim to recalibrate the distribution of the generated fake features. Deep features typically have higher dimensions, making the feature distribution clearer and making it easier to calibrate the feature distribution. The specific operational steps are as follows:

Calculate the mean and covariance of the input features. We feed the generated fake features into the distribution calibration module, where we use Equation 7 and 8 to calculate their mean and covariance, respectively. The mean is computed along the first dimension (sample dimension) of the features, while the covariance matrix represents the correlation and dispersion levels among the input features.

$$mean(x) = \frac{1}{n}\sum_{i=0}^{n} x_i \tag{7}$$

$$Cov(x) = \frac{\sum_{i=1}^{n} x_i - \bar{x}}{n-1} \tag{8}$$

Calculate the calibration matrix A. The function of calibration matrix A is establishing the relationship between the covariance of fake features and target features, ensuring that the calibrated statistical distribution (mean and covariance) matches that of the target features. The computation of the calibration matrix A can be represented as follows:

$$A = Cov(y) \cdot Cov(x)^T \tag{9}$$

y is the support feature, x is the generated fake feature.

Calibrated mean and covariance. We utilize the calibration matrix A to adjust the mean and covariance of fake features. The process of calibrating the mean and covariance are illustrated in Equation 10 and Equation 11:

$$Calibrated\_mean(x, y) = mean(y) + A \cdot (mean(y) - mean(x)) \tag{10}$$

$$Calibrated\_Cov(x) = A \cdot Cov(x) \cdot A^T \tag{11}$$

Here, A represents the calibration matrix, $Calibrated\_mean$ and $Calibrated\_Cov$ are the mean and covariance matrices after calibration. In this context, y represents the support features, and x represents the generated fake features. Specifically, first, this process involves calculating the deviation between the fake feature mean and the target mean. Subsequently, the calibration matrix A is multiplied by the mean deviation to adjust the difference between vector means. Finally, the adjusted mean deviation is added to the target mean, resulting in the calibrated mean $(i.e., Calibrated\_mean)$.

Generate calibrated features. First, we obtain a matrix B whose distribution is centered at zero mean by subtracting its mean vector from the fake feature. Then, we utilize the einsum function to perform matrix multiplication between matrix B and the input covariance matrix Cov(x), resulting in a covariance-transformed feature matrix, referred to as $data$. Finally, the calibration matrix A is

applied to the feature matrix $data$ and the $Calibrated\_mean$ vector is added to the result to generate calibrated $data$. The process can be represented as follows:

$$B = x_i - mean(x_i) \tag{12}$$

$$data = A \cdot einsum(cov(x), B) + Calibrated\_mean \tag{13}$$

In general, the distribution calibration module relies on the statistical distribution of support features. It remaps and adjust the generated feature distribution to match the target feature distribution through the linear transformation of the calibration matrix.

4. EXPERIMENTS AND ANALYSIS

To begin with, Sections 4.1 and 4.2 introduce the datasets utilized by the model and provide details regarding the experiments. Subsequently, Section 4.3 carries out a comparative analysis of various state-of-the-art methods on two distinct datasets. Finally, in Sections 4.4 to 4.6, a series of ablation analysis of the method proposed in this paper is conducted.

4.1. Datasets

The effectiveness of the CFSD model is evaluated using two widely used public datasets: PASCAL VOC (07+12) and the MS COCO dataset.

PASCAL VOC is a commonly used public dataset in the field of object detection, comprising 20 categories. In this study, the VOC 2007 train set and VOC 2012 train set are utilized for model training, and the model is tested on the VOC 2007 test set. The evaluation metric employed is novel category average precision (nAP0.5) with a threshold of 0.5. The 20 categories of the dataset are split into two parts: 15 base categories and 5 novel categories. The base and novel are randomly selected, with no overlap (i.e., Base ∩ Novel=∅). This study evaluates three different groups of Base/Novel samples. The novel category is selected in the same manner as Hu et al. [38]. There are three different novel categories: {“bird”, “bus”, “cow”, “motorbike”, “sofa”}, {"areophane", "bottle", "cow", "horse", "sofa"}, {"boat", "cat", "motorbike", "sheep", "sofa"}. During basic training, only the annotations of the base category are provided. For the few-shot fine-tuning stage, the model is trained using k images with annotated bounding boxes for each category, where k takes values of 1/2/3/5/10.

The MS COCO dataset comprises 80 object categories, with the same 20 categories as Pascal VOC designated as novel categories. The remaining 60 categories in the MS COCO dataset are utilized as basic categories. The process of constructing the few-shot dataset is similar to the Pascal VOC dataset, with k set to 10/30. We employ Train 2017 for training and Val 2017 for evaluation.

4.2. Implementation Details

We implement our method using MMDetection. In training, we adopt ResNet101 pre-trained on ImageNet-1K as the feature extractor and use SGD as the optimizer.

The experimental environment is based on a Linux system server with GCC version 6.1, and training is performed on a single GPU of NVIDIA A100 with 40G memory, and batch_size is set to 8. In the basic training stage of the model on the Pascal VOC dataset, the model was trained for 18000 iterations, and the initial learning rate of the model was 5×10^{-3}. The learning rate decays by 1×10^{-4} at 12000 and 16000 respectively. The learning rate remains unchanged during the few-shot fine-tuning phase. When 5×10^{-3}, the model iterates {400, 800, 1200, 1600, 2000} respectively. In the basic training stage of the model on the MS COCO dataset, the model was trained for 110000 iterations, and the initial learning rate was 125×10^{-5}. At the 85000 and 100000 iterations, the

learning rate decays by 1×10^{-4} respectively. In the few-shot fine-tuning stage, the learning rate is 5×10^{-3}. When $k = \{10, 30\}$, the model iterates {10000, 20000} respectively.

4.3. Comparison with State-of-the-Art Methods

In this section, we conduct several experiments to evaluate our proposed method. We utilize the Novel Class Average Precision (nAP0.5) with a threshold of 0.5 to evaluate the detection performance of the model and compare it with other excellent models. Finally, we perform ablation analysis for generative adversarial modules and distribution calibration.

4.3.1. Experimental Analysis of PASCAL VOC Dataset:

Table 1. Few-shot performance of the detection model on the Pascal VOC dataset. We evaluate the performance of three different sets of new classes under nAP0.5. In the results, * denotes the outcome obtained by re-running with the same parameters as ours.

Methods/Shots	Novel Set 1					Novel Set 2					Novel Set 3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
TFA w/ cos [33]	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
MPSR [34]	41.7	-	51.4	55.2	61.8	24.4	-	39.2	39.9	47.8	35.6	-	42.3	48.0	49.7
HallucFsDet[35]	47.0	44.9	46.5	54.7	54.7	26.3	31.8	37.4	37.4	41.2	40.4	42.1	43.3	51.4	49.6
Retentive [37]	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
SRR-FSD [36]	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
DC-Net [38]	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
FsDetView [39]	24.3	36.5	44.9	52.0	59.2	20.5	27.5	33.1	40.9	47.1	22.4	33.0	37.8	43.9	51.5
Meta-DETR[25]	35.1	49.0	53.2	57.4	62.0	27.9	32.3	38.4	43.2	51.8	34.9	41.8	47.1	54.1	58.2
MRSN [41]	47.6	48.6	57.8	61.9	62.6	31.2	38.3	46.7	47.1	50.6	35.5	30.9	45.6	54.4	57.4
CKPC [42]	35.1	45.5	48.2	52.3	60.2	25.2	32.4	39.5	44.3	49.2	25.2	35.8	40.9	48.0	54.6
VFA [43]*	48.1	57.4	61.6	64.2	64.0	37.1	46.7	48.3	50.8	51.2	43.1	51.1	54.5	57.7	59.0
CFSD (Ours)	48.6	58.2	62.3	64.9	65.4	38.6	46.8	49.8	51.6	52.0	43.4	52.8	55.2	59.7	59.5

In Table 1, we compare the performance of our method with other few-shot object detection approaches. Our model achieved the best results among 15 indicators, with an average increase of 1%for each indicator. Notably, it becomes apparent that as the number of samples for detector train increases, the test results also improve. This phenomenon is attributed to the generative adversarial module capturing more diverse feature information as the number of train samples grows. Additionally, this module turns a few-shot problem into a multi-sample problem, such as augmenting from 1-shot to 2-shot, augmenting from 10-shot to 20-shot, thereby providing the detection head with more samples to more accurately determine the decision boundary.

4.3.2. Experimental Analysis of MS COCO Dataset.

Table 2. Few-shot performance of the detection model on the MS COCO dataset.

Methods		10-shot			30-shot	
	nAP 0.50:0.95	nAP 0.50	nAP 0.75	nAP 0.50:0.95	nAP 0.50	nAP 0.75
TFA w/ cos [33]	-	10.0	13.7	-	9.3	13.4
MPSR [34]	9.8	17.9	9.7	14.1	25.4	14.2
Retentive [37]	-	10.5	-	-	13.8	-
SRR-FSD [36]	11.3	23.0	9.8	14.7	29.2	13.5
DC-Net [38]	12.8	23.4	11.2	18.6	32.6	17.5
FsDetView [39]	13.4	30.6	9.1	17.1	35.2	14.7
Norm-VAE [40]	-	18.7	17.8	-	22.5	22.4
CKPC [42]	16.6	34.4	17.2	19.9	38.1	19.7
VFA [43]	16.2	35.3	11.4	18.9	38.6	15.8
CFSD (Ours)	16.5	36.8	12.4	19.5	40.3	17.5

We evaluate the results for 10/30-shot evaluations on the challenging MS COCO dataset. The evaluation results for the novel categories are presented in Table 2. The model demonstrates a notable improvement of over 1% on the commonly used nAP0.50 evaluation index. Additionally, we report two more stringent indicators, namely nAP0.50:0.95 and nAP0.75. Compared to the robust baseline (VFA), our method exhibits increase of 0.3% and 0.6% for the nAP0.50:0.95 index, and 1% and 1.7% for the nAP0.75 index, respectively. Although the model's performance on the nAP0.50:0.95 and nAP0.75 indicators doesn't reach the best results, it still surpasses many excellent methods.

4.4. Ablation Experiment

To verify the effectiveness of our method and demonstrate its superior performance, we conducted several ablation experiments and given comparative experimental analysis. We employed 5 novel categories of Novel Set 1 as training samples for evaluation. The ablation experiment evaluates the effectiveness of each module and provide data support for technology upgrades and improvements.

Table 3. Ablation Experiment. Effectiveness of the Gan module and Calibration module

Methods	GAN	Calibration	Novel Set 1 nAP0.50				
			1-shot	2-shot	3-shot	5-shot	10-shot
VFA [43]			48.1	57.4	61.6	64.2	64.0
VFA (without VAE)			45.6	55.3	57.6	60.5	60.7
VFA (without VAE)	√		47.8	57.4	60.9	63.7	64.6
VFA (without VAE)	√	√	48.6	58.2	62.3	64.9	65.4

Table 3 shows the results of the ablation experiments. Since VAE is a generative model and conflicts with the approach in this paper, we excluded the VAE module from the strong baseline VFA. It is evident that after removing VAE, the accuracy of the VFA model significantly decreases. Subsequently, by incorporating the Generative Adversarial Network (GAN) module only, the model achieves a 2% to 3% improvement. but the results do not surpass the original method. This is because the statistical distribution of the fake features generated by GAN are biased toward the base classes. Thereby, it interferes with the decision boundary of the novel classes in the feature space cause by limited training samples. When both the generative module and the distribution calibration module are used simultaneously, the model's performance surpasses that of the source model (VFA), with an average improvement of 1% to 2% for each metric. This improvement is attributed to the distribution calibration module correcting the intra-class statistical distribution of fake features, making the classification decision boundary clearer.

4.5. How the Distribution Calibration Module Affects The Distribution Dispersion of Fake Features

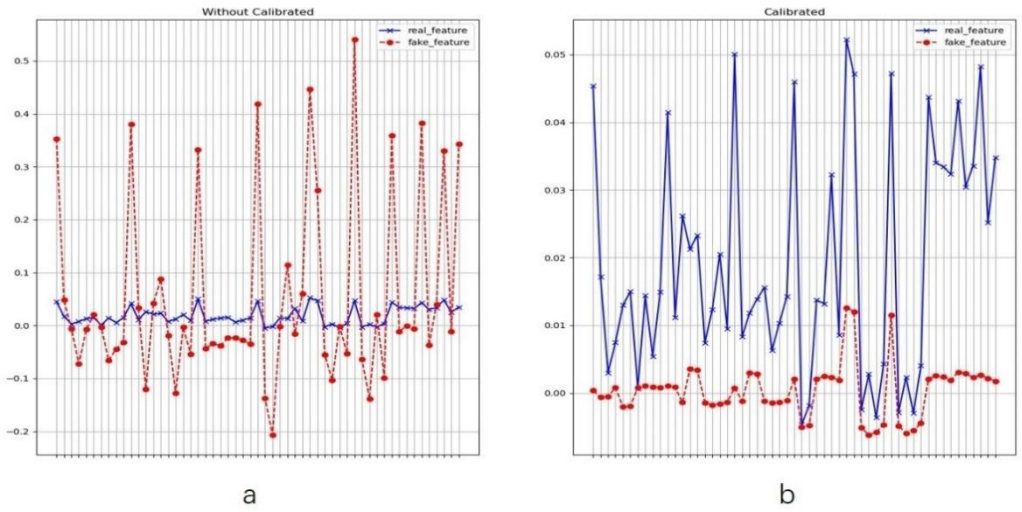

Figure 3. The impact of the distribution calibration module on the discrete degree of fake feature distribution

In feature space, the covariance of features is proportional to their degree of dispersion. A larger covariance indicates a more discrete feature distribution, and an excessively discrete distribution can result in a blurry classification boundary. In Figure 3, we analyze how the dispersion of the fake feature (red dashed line) changes under the influence of the distributed calibration module. In Figure a, it illustrates the scenario where the model did not use the distribution calibration module, and the covariance of fake features ranges from -0.2 to 0.6. The covariance of real features (blue line) is approximately -0.01 to 0.06. It is evident that the dispersion of the fake feature distribution is much higher than that of the real feature. In Figure b, after the features are corrected by the distribution calibration module, the degree of dispersion of the fake features is reduced to between -0.01 and 0.02, which is lower than the real features (note that the values of the Y-axis are different in Figure a and Figure b). The above data fully demonstrate that the distributed calibration module can correct the fuzzy features of the classification boundary and reduce the dispersion of fake feature distribution.

4.6. Analysis of Generated Adversarial Losses

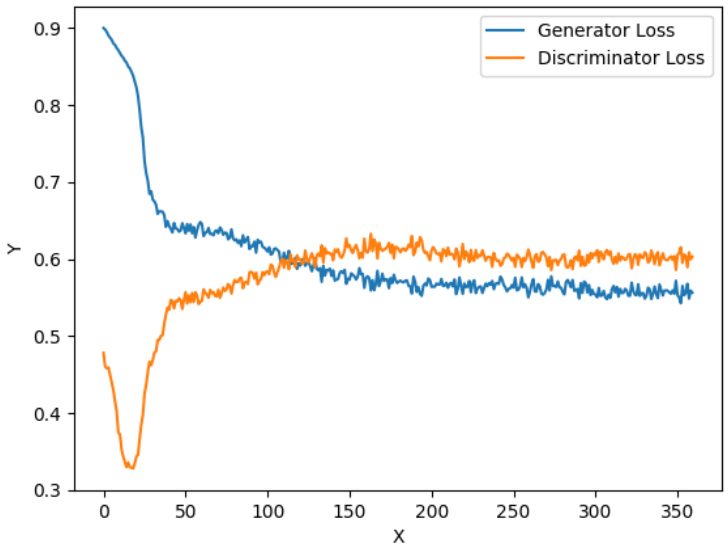

Figure 4. Generate adversarial loss curve graph

Fig. 4 illustrates the generator loss and discriminator loss. The adversarial interaction between these two losses initiates a dynamic competition between the generator and the discriminator. This iterative process continues until the generated features attain a level of realism where the discriminator can no longer distinguish between true and false features. Upon reaching a state of Nash equilibrium, the fake features closely approximate the characteristics of real features, and achieve the desired quality.

4.7. Visual Analysis of the Effect of Distribution Calibration

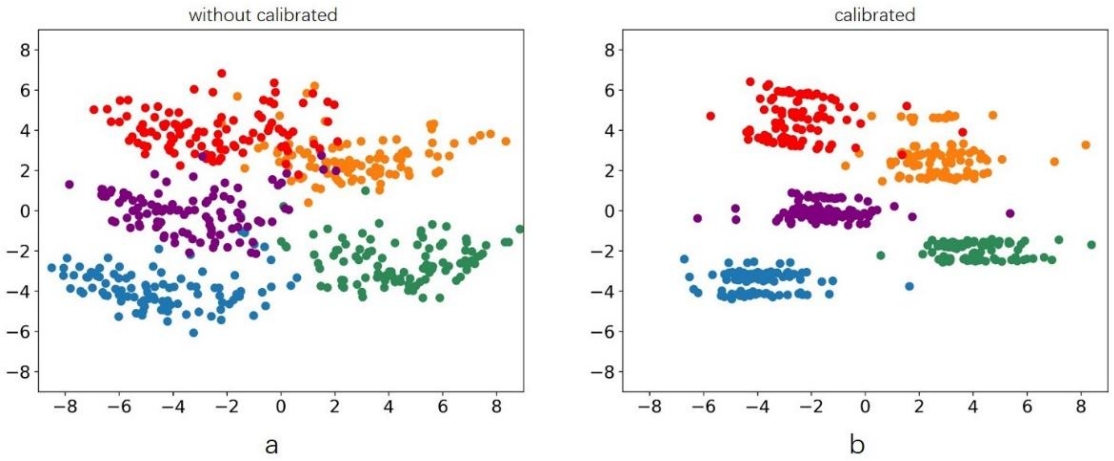

Figure 5. Effect of distribution calibration on fake features in feature space

We conducted a visual analysis of the 10-shot results for the 5 categories in Novel Set 1, selecting 100 fake features for each category. Each point in Figure 5 corresponds to a fake feature, with different colors representing different categories. In Figure a, we observe the test results without using distribution calibration. It is evident that the decision boundaries between different categories are not clear, particularly the pronounced conflict between red and orange. In Figure b, we observe the test results after applying distribution calibration, revealing distinct boundaries between different categories. While there is a minor conflict between the red and orange parts, the classification effectiveness is significantly improved compared to Figure a.

5. VISUALIZATION

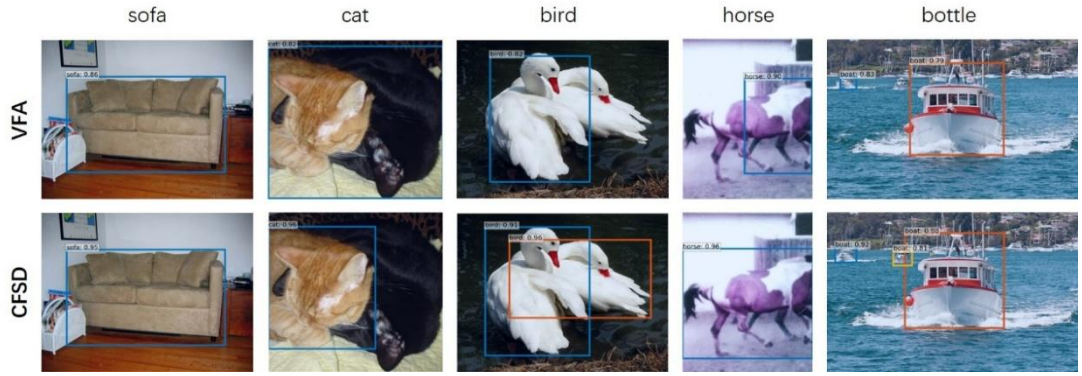

Figure 6. Visualization of detection results compared with strong baselines. The model is trained on Pascal VOC Novel Set 1, 2, 3 and tested on the VOC07 test set

The Pascal VOC dataset encompasses various practical application scenarios, providing a thorough assessment of the model's generalization capability. Figure 6 illustrates the visualization of results obtained by the CFSD method proposed in this paper and the robust baseline method VAF on the test set. Samples were chosen from three distinct sets of novel categories, and the white portions in the figure indicate the detection confidence of the target category. Clearly, under the influence of CFSD,

the detection confidence for five different categories has significantly improved. Particularly in the results of the third column, CFSD successfully detected all targets and accurately identified overlapping areas even when the targets intersected. Additionally, it was observed that the blue detection box of CFSD exhibited greater accuracy than the blue detection box of VAF. In the fifth column, CFSD detected small targets that were overlooked by the baseline method, achieving a detection confidence of 81%. This demonstrates that the method in this paper can effectively enhance the model's feature representation of small targets.

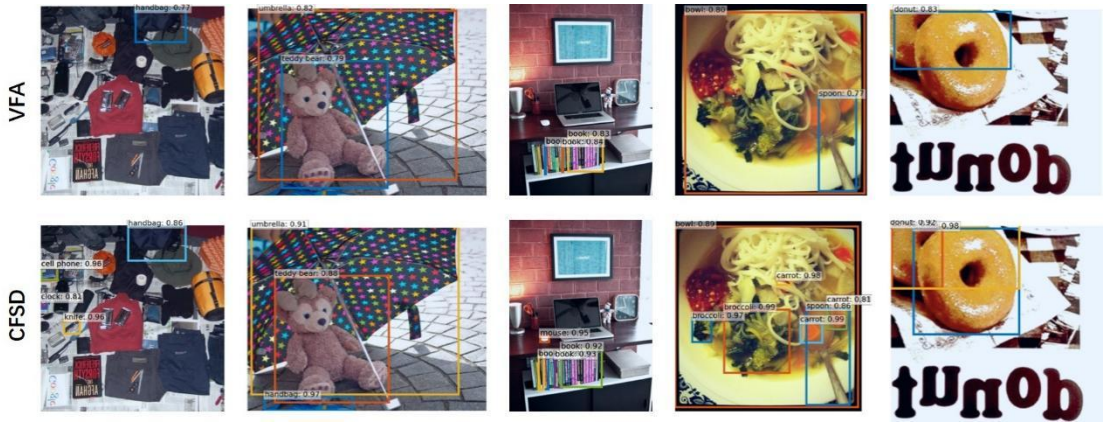

Figure 7. Visualization of the detection results of the CFSD model on MS COCO dataset

We also present the model's performance on the MS COCO dataset. The challenge level of COCO exceeds that of VOC due to its inclusion of 80 categories and a significant number of small objects, which posing a more formidable task for the detector. Examining the results in the first and third columns in Figure 7, it is evident that the CFSD model outperforms the VFA model in the detection of small objects. In the second and fifth columns, the model demonstrates the ability to accurately locate object even in the presence of occlusions. Moreover, the CFSD model exhibits significantly higher confidence in object detection compared to the robust baseline model. The experiment results on COCO conclusively demonstrate that feature augmentation and distribution calibration contribute to improving the model's discriminative representation of novel categories.

6. CONCLUSIONS

This paper introduces a generative adversarial module to address the issue of overfitting in the detection head and classifier caused by insufficient samples. Additionally, the statistical distribution of fake features generated by the generative adversarial module tends to be biased towards the base category due to limited samples. To mitigate this, we propose a distribution calibration method. This approach adjusts the statistical distribution of fake features to align with the distribution of real categories, aiding the classifier in establishing accurate decision boundaries. The proposed method is evaluated on the Pascal VOC and MS COCO datasets, demonstrating its effectiveness in overcoming these challenges and outperforming existing state-of-the-art FSOD models.

ACKNOWLEDGEMENTS

The work was funded by Foundation of Henan Province philosophy social sciences with grant number 2025JYQS1121, and Double First Class Construction Project of Henan Polytechnic University with grant number GCCYJKT202515.

REFERENCES

[1] Dong X, Zheng L, Ma F, et al. Few-example object detection with model communication [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(7): 1641-1654.

[2] Chen H, Wang Y, Wang G, et al. Lstd: A low-shot transfer detector for object detection [C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

[3] Wang Y X, Ramanan D, Hebert M. Meta-learning to detect rare objects [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9925-9934.

[4] Kang B, Liu Z, Wang X, et al. Few-shot object detection via feature reweighting [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 8420-8429.

[5] Yan X, Chen Z, Xu A, et al. Meta r-cnn: Towards general solver for instance-level low-shot learning [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9577-9586.

[6] Ni R, Goldblum M, Sharaf A, et al. Data augmentation for meta-learning [C]//International Conference on Machine Learning. PMLR, 2021: 8152-8161.

[7] Rajendran J, Irpan A, Jang E. Meta-learning requires meta-augmentation [J]. Advances in Neural Information Processing Systems, 2020, 33: 5705-5715.

[8] Pei Z, Jiang H, Li X, et al. Data augmentation for rolling bearing fault diagnosis using an enhanced few-shot Wasserstein auto-encoder with meta-learning [J]. Measurement Science and Technology, 2021, 32(8): 084007.

[9] Zhang J, Zhao C, Ni B, et al. Variational few-shot learning [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1685-1694.

[10] Xu J, Le H, Huang M, et al. Variational feature disentangling for fine-grained few-shot classification [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 8812-8821.

[11] Lin X, Duan Y, Dong Q, et al. Deep variational metric learning [C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 689-704.

[12] Gao H, Shou Z, Zareian A, et al. Low-shot learning via covariance-preserving adversarial augmentation networks [J]. Advances in Neural Information Processing Systems, 2018, 31.

[13] Wang Y X, Girshick R, Hebert M, et al. Low-shot learning from imaginary data [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7278-7286.

[14] Zhang R, Che T, Ghahramani Z, et al. Metagan: An adversarial approach to few-shot learning [J]. Advances in neural information processing systems, 2018, 31.

[15] Li K, Zhang Y, Li K, et al. Adversarial feature hallucination networks for few-shot learning [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 13470-13479.

[16] Zhao J, Mathieu M, LeCun Y. Energy-based generative adversarial network [J]. arXiv preprint arXiv:1609.03126, 2016.

[17] Tan C, Sun F, Kong T, et al. A survey on deep transfer learning [C]//Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27. Springer International Publishing, 2018: 270-279.

[18] Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning [J]. Proceedings of the IEEE, 2020, 109(1): 43-76.

[19] Qiao L, Zhao Y, Li Z, et al. DeFRCN: Decoupled Faster R-CNN for few-shot object detection [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 8681-8690.

[20] Zhu C, Liang J, Zhou F. Transfer learning-based YOLOv3 model for road dense object detection [J]. Journal of Electronic Imaging, 2023, 32(6): 062505-062505.

[21] Hospedales T, Antoniou A, Micaelli P, et al. Meta-learning in neural networks: A survey [J]. IEEE transactions on pattern analysis and machine intelligence, 2021, 44(9): 5149-5169.

[22] Beck J, Vuorio R, Liu E Z, et al. A survey of meta-reinforcement learning [J]. arXiv preprint arXiv:2301.08028, 2023.

[23] Huisman M, Van Rijn J N, Plaat A. A survey of deep meta-learning [J]. Artificial Intelligence Review, 2021, 54(6): 4483-4541.

[24] Fu K, Zhang T, Zhang Y, et al. Meta-SSD: Towards fast adaptation for few-shot object detection with meta-learning [J]. Ieee Access, 2019, 7: 77597-77606.

[25] Zhang G, Luo Z, Cui K, et al. Meta-DETR: Image-level few-shot detection with inter-class correlation exploitation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[26] Liu J, Chao F, Lin C M. Task augmentation by rotating for meta-learning [J]. arXiv preprint arXiv:2003.00804, 2020.

[27] Trabucco B, Doherty K, Gurinas M, et al. Effective data augmentation with diffusion models [J]. arXiv preprint arXiv:2302.07944, 2023.

[28] Cheng M, Wang H, Long Y. Meta-learning-based incremental few-shot object detection [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 32(4): 2158-2169.

[29] Han J, Ren Y, Ding J, et al. Few-shot object detection via variational feature aggregation [J]. arXiv preprint arXiv:2301.13411, 2023.

[30] Zhang J, Zhang X, Wang Z. Task encoding with distribution calibration for few-shot learning [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(9): 6240-6252.

[31] Park S J, Han S, Baek J W, et al. Meta variance transfer: Learning to augment from the others [C]//International Conference on Machine Learning. PMLR, 2020: 7510-7520.

[32] Yang S, Liu L, Xu M. Free lunch for few-shot learning: Distribution calibration [J]. arXiv preprint arXiv:2101.06395, 2021.

[33] Wang X, Huang T E, Darrell T, et al. Frustratingly simple few-shot object detection [J]. arXiv preprint arXiv:2003.06957, 2020.

[34] Wu J, Liu S, Huang D, et al. multi-scale positive sample refinement for few-shot object detection [C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. Springer International Publishing, 2020: 456-472.

[35] Zhang W, Wang Y X. Hallucination improves few-shot object detection [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13008-13017.

[36] Zhu C, Chen F, Ahmed U, et al. Semantic relation reasoning for shot-stable few-shot object detection [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 8782-8791.

[37] Fan Z, Ma Y, Li Z, et al. Generalized few-shot object detection without forgetting [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 4527-4536.

[38] Hu H, Bai S, Li A, et al. Dense relation distillation with context-aware aggregation for few-shot object detection [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 10185-10194.

[39] Lee H, Lee M, Kwak N. Few-shot object detection by attending to per-sample-prototype [C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022: 2445-2454.

[40] Xu J, Le H, Samaras D. Generating Features with Increased Crop-related Diversity for Few-Shot Object Detection [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 19713-19722.

[41] Ma T, Bi M, Zhang J, et al. Mutually Reinforcing Structure with Proposal Contrastive Consistency for Few-Shot Object Detection [C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 400-416.

[42] Chen C, Yang X, Zhang J, et al. Category knowledge-guided parameter calibration for few-shot object detection [J]. IEEE Transactions on Image Processing, 2023, 32: 1092-1107.

[43] Han J, Ren Y, Ding J, et al. Few-shot object detection via variational feature aggregation [J]. arXiv preprint arXiv:2301.13411, 2023.

[44] Shao X, Zhang W. Spatchgan: A statistical feature based discriminator for unsupervised image-to-image translation [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6546-6555.

[45] Ding L, Tang H, Bruzzone L. LANet: Local attention embedding to improve the semantic segmentation of remote sensing images [J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 59(1): 426-435.