TYPE Original Research
PUBLISHED 01 May 2023
DOI 10.3389/fmars.2023.1151112

Check for updates

OPEN ACCESS

EDITED BY
Hongsheng Bi,
University of Maryland, College Park,
United States

REVIEWED BY
Ercan Avşar,
Technical University of Denmark, Denmark
Leonardo Bobadilla,
Florida International University,
United States

*CORRESPONDENCE
Zhibin Yu
✉ yuzhibin@ouc.edu.cn

RECEIVED 25 January 2023
ACCEPTED 11 April 2023
PUBLISHED 01 May 2023

CITATION
Han L, Zhai J, Yu Z and Zheng B (2023) See you somewhere in the ocean: few-shot domain adaptive underwater object detection.
Front. Mar. Sci. 10:1151112.
doi: 10.3389/fmars.2023.1151112

See you somewhere in the ocean: few-shot domain adaptive underwater object detection

Lu Han[1,2], JiPing Zhai[1,2], Zhibin Yu[1,2]* and Bing Zheng[1,2]

[1]Department of Electronic Engineering, College of Information Science and Engineering, Ocean University of China, Qingdao, Shandong, China, [2]Sanya Oceanographic Institution, Ocean University of China, Sanya, China

The current data-driven underwater object detection methods have significantly progressed. However, there are millions of marine creatures in the oceans, and collecting a corresponding database for each species for similar tasks (such as object detection)is expensive. Besides, marine environments are more complex than in-air cases. Water quality, illuminations, and seafloor topography may lead to domain shifting with visual instability features of underwater objects. To tackle these problems, we propose a few-shot adaptive object detection framework with a novel two-stage training approach and a lightweight feature correction module to accommodate both image-level and instance-level domain shifting on multiple datasets. Our method can be trained in a source domain and quickly adapt to an unfamiliar target domain with only a few labeled samples. Extensive experimental results have demonstrated the knowledge transfer capability of the proposed method in detecting two similar marine species. The code will be available at: https://github.com/roadhan/FSCW

KEYWORDS

computer vision, underwater object detection, domain adaptive, few shot, deep learning

1 Introduction

In recent years, with the development of deep learning technology and the deterioration of the marine ecological environment, underwater optical object detection has attracted more and more attention. However, many problems still need to be solved in underwater object detection. On the one hand, the underwater environment is complex and changeable. Affected by the scattering and absorption of the water medium, the quality of the images is usually poor (Fu et al., 2023). These underwater factors would inevitably involve inconsistent visual features. On the other hand, underwater images are challenging to collect and have limited reusability. Suppose we need more samples to boost a deep-learning model to handle a detection task. Generally, a common method is to use another large-scale dataset (e.g., Microsoft Common Objects in Context(MSCOCO) dataset Lin et al. (2014)) to boost the model and finetune the model with limited samples with new categories (Cai et al., 2022). This method can be particularly helpful for new dataset tasks if the large-scale dataset contains similar target categories (Zhu et al., 2021a) (e.g., the experience of motorbike detection can help bicycle detection in another task). However, there are domain shifts between different datasets due to differences in shots, environments, and objects themselves (Li et al., 2022a; Yu et al., 2022). These domain shifts prevent us from fully exploiting prior knowledge on large datasets. Therefore, in-air adaptive object detection algorithms are designed to solve such problems. Different underwater optical characteristics can also easily cause domain shifts (Liu et al., 2020), resulting in hue changes and discrepancies in visual features. Moreover, due to changes in the ecological environment of the new waters, similar species may also have different appearance characteristics. Therefore, under these domain shifts, datasets collected in one water body are unlikely to help detection tasks in another water environment.

Similar to in-air domain adaptive object detection (Wang et al., 2019), we can divide underwater domain shift into image-level domain shift and instance domain shift. Image-level shift refers to the shift of the image in terms of style, brightness, etc. As shown in Figure 1, we attribute water transparency and chromatic aberration to image-level shift underwater. Instance-level shift refers to the shift of the target in appearance and size. We group organisms of the same family or genus but different species as instance-level shifts underwater. Any domain shift will have a significant performance degradation on the underwater detection network. The green bounding box represents the undetected target. Regarding results in Figure 1, the detection network can hardly work well under image-level and instance-level shifts.

Unsupervised domain-adaptive object detection based on deep learning is generally considered a solution to this kind of problem (Chen et al., 2018; Saito et al., 2019; Shen et al., 2019). However, the current domain-adaptive object detection algorithms have several apparent flaws. First, these methods always need a large amount of target domain data for training (Wang et al., 2019), which is difficult to obtain in underwater scenes. Second, due to the algal blooms or river floods at different times, the environmental conditions of the offshore and river outlets may change unexpectedly and cause a changeable aquatic background.

Although many existing few-shot object detection methods can work with a few data, their feature extraction ability on the new domain will be significantly affected by the changeable aquatic background. This is because most existing few-shot object detection considered shared weights or a separately trained feature extraction module to extract the feature map of the new class. Since the model has yet to see the new domain, the feature extraction ability on the new domain would be insufficient (Li et al., 2022d). On the other hand, most domain-adaptive methods can adapt to a new domain with sufficient retraining on the source domain and target domain data (Wang et al., 2019). However, such methods usually need a large amount of target domain data. The lengthy retraining time also hinders further applications on underwater vision.

Inspired by the theory of few-shot learning (Kang et al., 2019) and transfer learning (Sun et al., 2021), we propose a fast few-shot domain-adaptive algorithm to tackle the challenge of underwater cross-domain object detection. Our contributions can be summarized as follows: 1) Aiming at the problem of insufficient ability of the backbone to extract features, as shown in Figure 2, we fused the two-branch algorithm into a single-branch object detection algorithm with a channel-level feature correction module to solve this problem. 2) Many existing domain adaptation algorithms need a long time to adjust to a new domain. We propose a two-stage domain adaptation training strategy, which only takes a short time to adapt to the new target domain. 3) We conduct exhaustive experiments on two datasets, demonstrating that our algorithm performs excellently on few-shot domain adaptation problems. Compared to other domain adaptation algorithms, our algorithm has two key advantages:

1) **Boosting the model with limited data.** Compared with unsupervised domain adaptation (UDA) object detection, which requires many unlabeled samples, our model only needs a small number of labeled samples to complete the training and achieve excellent performance during the target domain adaptation.
2) **Adapting new tasks with less time.** When our model encounters unfamiliar environments, it no longer needs to be trained on both the source and target domain data simultaneously. Instead, it only needs to be fine-tuned on a small number of labeled target domain data sets, which reduces the adaptation time.

2 Related work

General Object Detection refers to finding the object we need from the image and giving an accurate mark frame and category (Li et al., 2022a). Current deep learning-based object detection can be divided into two architectures: one-stage and two-stage methods. The two-stage methods are mainly based on the region convolutional neural network (R-CNN) series. They use a convolutional neural network (CNN) to generate region proposals where objects may exist and perform further category prediction and bounding box regression in the detection head module. The one-stage methods perform end-to-end bounding box regression and category prediction through the neural network. The one-stage methods include You Only Look Once (YOLO) (Redmon et al., 2016; Zhu et al., 2021b), RetinaNet (Lin et al., 2017b), etc. Usually, two-stage methods outperform one-stage methods in accuracy, but they have poorer inference speeds. Both two architectures require large datasets for training. Considering the real-time requirements of underwater object detection, we use YOLOv5 (Zhu et al., 2021b) as the baseline in this paper

Underwater Object Detection is a particular branch of object detection. Compared with general object detection tasks, underwater images often have problems such as blurring, color

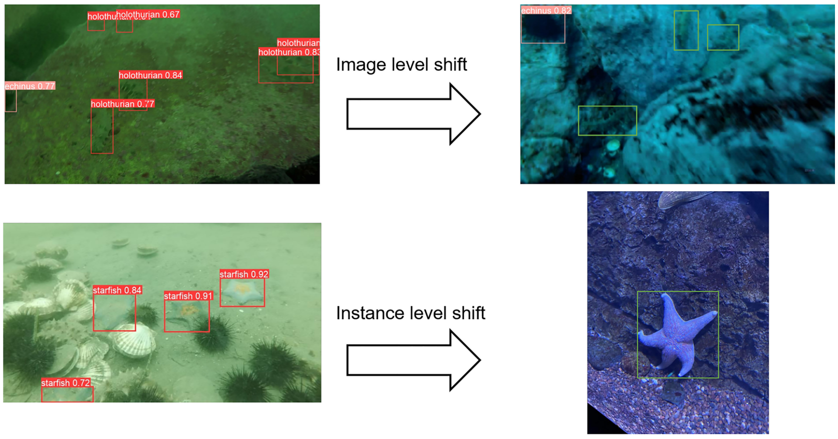

FIGURE 1
Two different underwater domain shifts and cross-domain performance degradation of detectors.

shifting, and costly data collection. To tackle these problems, (Lin et al., 2020) proposed an augmentation method called the region of interest mix-up (RoIMix), by fusing the proposed regions of different images to enhance the generalization of the detection network. (Fan et al., 2020a) proposed an underwater detection framework with feature enhancement and anchor refinement, which improves the ability of the detector to deal with underwater images of different scales. (Liang and Song, 2022) applying Self-Attention modules to the region of interest (RoI) features to improve underwater detector performance. However, the underwater objection detection methods often have to be deployed in an unseen underwater environment, which can lead to a domain shift. Unfortunately, the current underwater object detection algorithms have not yet considered the problem of adapting to different waters.

Domain Adaptation refers to reducing domain shift by training neural networks on source and target domain datasets. The current domain adaptive object detection is mainly based on unsupervised domain adaptation. According to the domain adaptation theory (Ganin et al., 2016), when performing neural network domain adaptation, the features extracted by the backbone must have domain invariant properties to adapt to a new domain. Ganin et al. 2016 used a gradient reversal layer with a domain classifier to constrain the backbone to extract features without domain shift to achieve this goal. This method is called domain adversarial training, which is still adopted by most domain adaptation methods. (Chen et al., 2018) divides domain shift into image-level and instance-level domain shift, and two adaptive components are designed to adapt to these two domain shifts, respectively. (Saito et al., 2019) designed a weak alignment model using adversarial alignment loss to address domain variance. (Kiran et al., 2022) proposes the domain transfer module (DTM) to transform the source image according to different target domain images, enabling the network to avoid catastrophic forgetting when performing multi-domain adaptation. (Li et al., 2022b) proposed a novel semantic conditional adaptation framework to reduce the cross-domain misclassification problem. The above works only focus on domain adaptation under large unsupervised samples and do not consider the problems encountered in few-shot domain adaptation. In the case of only a small number of samples, labeling samples do not add too much labor overhead. (Wang et al., 2019) considers the domain adaptation problem under the condition of small sample labeling. He proposed a two-layer module to adapt to the domain adaptive object detection problem under limited loose labeling. Loose labeling means that only part of each image is labeled, and more images are used to improve the target information of labeling. This method is promising for cases when image acquisition is easy but labeling is complex. Nevertheless, the reverse more or less applies in underwater object detection. Collecting underwater data is always expensive and time-consuming, but labeling objects is relatively easy. Unlike other domain adaptation methods, our model can quickly adapt to the target domain when there are only a few labeled samples in the target domain.

Few-shot learning refers to learning new categories with limited data. In the field of object detection, methods can be divided into two main branches: dual-branch methods and single-branch methods (Köhler et al., 2021). The dual-branch methods are shown in Figure 3A, and an auxiliary feature extraction module is used to extract the feature vector of the support set image. Support set vectors are then channel-level interacted with query set vectors. (Kang et al., 2019) use a pre-trained backbone on the basis of YOLO to extract the support set feature vector which will reweight the query set vector. (Fan et al., 2020b) on the basis of Faster-RCNN, use the shared weight backbone to extract the support set feature vector to complete the reweighting step and use the multi-relation detector to classify the target. (Lee et al., 2022) propose a method to refine the support information through an attention mechanism among support data before aggregating the query and support data. The single-branch methods are shown in Figure 3B. The single-branch methods are mainly based on transfer learning. (Wang et al., 2020) used the transfer learning theory to unfreeze the bounding box regression and the classification layer of Faster R-CNN achieves excellent performance. Sun et al. proposed a method (Sun et al., 2021) by controlling the form of intersection over the union (IoU) output with the Faster R-CNN of the unfreezing region proposal network (RPN) and region of interest (ROI) pooling layers and

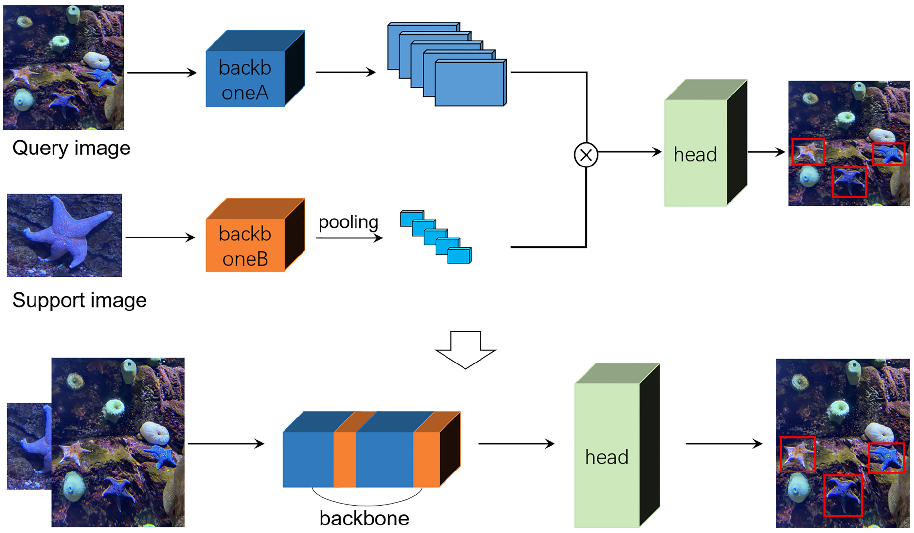

FIGURE 2
Improve the dual-branch structure to a single-branch structure that is more suitable for domain adaptation problems.

achieved the best performance at that time. Generally, the single-branch methods have only one backbone with fewer parameters and converge faster. Since the dual-branch methods considered meta-learning and more parameters, they can achieve better performance on few-shot learning. But their training speed is lower than single-branch cases. Since both two kinds of methods did not consider domain shifts, they will lead to a dramatic drop in performance when handling new samples from another domain. Our method combines the advantages of both approaches. To address this issue, we propose the feature correction module (FCM) (Figure 3C), which plays a similar role on the backbone B of dual-branch methods to enhance feature extraction ability with few samples. Furthermore, we use a two-stage fine-tuning method to make our model adjust itself to the features of the new domain.

3 Method

This section will briefly introduce our few-shot domain adaptive object detection algorithm. Due to the insufficient samples in the underwater target domain, the existing domain adaptation methods cannot achieve good results. The main reason for the poor performance in cross-domain object detection tasks is that the feature extraction ability of the backbone can hardly work in new domains (Li et al., 2022d). To solve this problem, we propose a solution. Firstly, we can overcome the overfitting problem of few-shot by introducing a two-stage training strategy. The proposed strategy can also reduce the need for repeated training on the source domain, shortening the time to adapt to the new domain. Secondly, by introducing a feature correction module, we

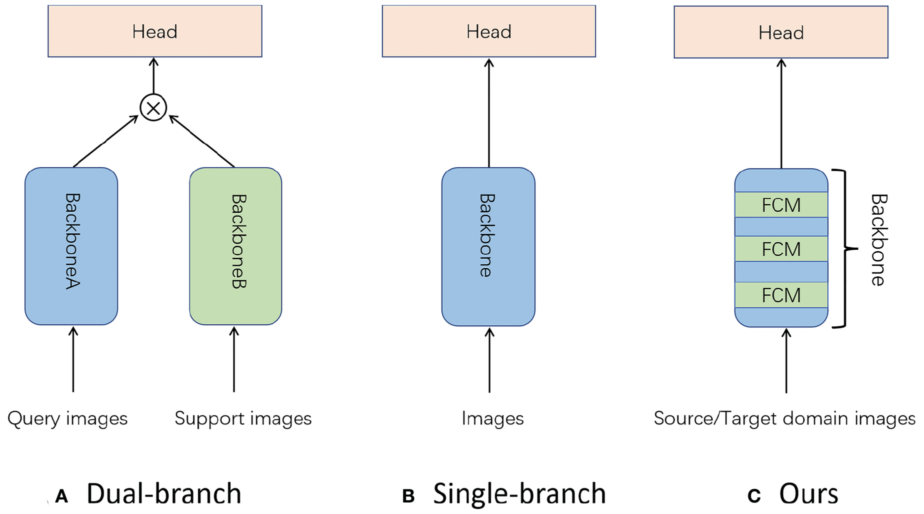

FIGURE 3
The comparison among two typical few-shot learning structures and ours.

further enhance the feature extraction ability of the backbone on new domains. Since the feature correction module only contains quite a few trainable parameters, it only takes a little for training. When only a few labeled samples are in the target domain, our method can quickly adapt to the target domain and achieve excellent performance.

3.1 Problem definition

We follow and extend the definition of "n-shot learning" given by (Kang et al., 2019). Suppose we have k images with labels in the source domain. We can define these images and labels in the source domain as $D_s = \{(X_{s1}, Y_{s1}), \ldots, (X_{sk}, Y_{sk})\}$. Similarly, we can define the images and labels in the target domain as $D_t = \{(X_{t1}, Y_{t1}), \ldots, (X_{tm}, Y_{tm})\}$. Since the target domain data is often less than the source domain, we have $k >> m$. Here D_s and D_t represent the source domain and target domain data, respectively. X and Y represent the images and the corresponding target labels. Let $num()$ denote the number of instances in a domain. In Kang et al.'s work (Kang et al., 2019), they defined n-shot ($num(X) = n$) as available samples (instances) in a domain. In the case of instance-level domain shift, the shape of the target will change significantly with the region. Thus, we followed this definition to evaluate instance-level domain shifts as $n - shot_{instance} = num(X_{t1}) + \ldots + num(X_{tm})$, where the $num()$ is the number of instances of an image. Since the main factor to cause image-level domain shifts is the environment (not the objects), we further define $n - shot_{image} = num'(X_t))$, where the $num'()$ means the number of images.

3.2 Two-stage fine-tuning method

Most existing few-shot learning approaches consider only adjusting the classification and the bounding Box regression header without changing the parameters of the backbone (Wang et al., 2020). Such an operation can correct new few-shot samples in a short time. However, the underwater domain shifts will also affect the backbone rather than the header. Inspired by Li et al.'s work (Li et al., 2022d), which proposed a two-stage fine-tuning strategy to correct a cross-domain classification task, we further extend the fine-tuning method to solve a cross-domain object detection problem.

Since different layers of the backbone network can extract different scales of features (Lin et al., 2017a), we focus on the domain correction of the backbone. Furthermore, to reduce the fine-tuning cost and accelerate the re-training speed, we insert some feature extraction modules (FCMs, please refer to Section 3.3 for detail) into the backbone and only update these feature extraction modules in the fine-tuning phase. As a result, our two-stage fine-tuning strategy can reduce the number of trainable parameters to solve the overfitting problem of few-shot. Our two-stage training method reduces the number of parameters by 42% compared to direct training YOLOv5, while our newly added FCM only increases the number of parameters by 0.00278%. The training method is shown in Figure 4. To overcome the underwater cross-domain challenge, our method includes two stages:

Base training: Our first stage is only performed on the source domain training dataset. In order to ensure a fair comparison, except for the modification of the network module we will mention in Section 3.3, the training hyperparameters remain the same as those of YOLOv5. We did not perform any hyperparameter tuning. The joint loss function is:

$$L_{total} = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{box} \tag{1}$$

where λ_1, λ_2 and λ_3 is the custom hyperparameter. Among them, both L_{cls} and L_{obj} using binary cross entropy (BCE) loss for classification and foreground detection, respectively:

$$L_{BCE} = -\frac{w}{N}\sum_{n=1}^{N}[y_n \cdot \log F(x)_{x\sim P_s(x)} + (1-y_n) \cdot \log(1 - F(x)_{x\sim Ps(x)})] \tag{2}$$

where w is a hyperparameter, x and y represents different images and labels. $P_{s/t}$ represents our network to obtain data from the source or target domain at different training stages. represents the number of samples. F represents the model. L_{box} uses CIoU loss (Zheng et al., 2020):

$$L_{CIoU} = IoU - \left(\frac{\rho^2(b^s, b^s_{gt})}{c^2} + \alpha v\right) \tag{3}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w}{h}\right)^2 \tag{4}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{5}$$

where ρ represents the Euclidean distance between b^s and b^s_{gt}, $b^{s/t}$ and $b^{s/t}_{gt}$ represents the detected bounding box and ground truth on the

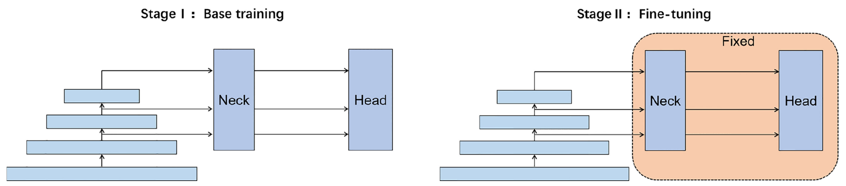

FIGURE 4
Our two-stage training approach.

source domain dataset or target domain dataset. *IoU* represents the intersection over the union.

Fine-tuning: Our second stage (fine-tuning) is performed on a small amount of labeled target domain data. In this stage, we freeze the neck and head modules of the detection network and only perform gradient updates on the backbone, the function is:

$$\frac{\partial \mathrm{L}_{total}}{\partial\, net_b^t} = \frac{\partial \mathrm{L}_{total}}{\partial\, net_h^t} \cdot \frac{\partial\, net_h^t}{\partial\, net_b^t} \tag{6}$$

$$W(net_h^t) \equiv W(net_h^s) \tag{7}$$

Among them, $net_h^{t/s}$ represents the neck and head network modules on the target domain dataset or source domain dataset, and net_b^t represents the backbone module on the target domain dataset. W represents the network weight.

Since the target domain dataset is adopted in the fine-tuning stage, our BCE loss and CIoU loss function are changed accordingly to:

$$L_{BCE} = -\frac{w}{N}\sum_{n=1}^{N}\left[y_n \cdot \log F(x)_{x\sim P_t(x)} + (1-y_n)\cdot \log\left(1 - F(x)_{x\sim P_t(x)}\right)\right] \tag{8}$$

$$L_{CIoU'} = IoU - \left(\frac{\rho^2(b^t, b^t_{gt})}{c^2} + \alpha v\right) \tag{9}$$

3.3 Lightweight feature correction module

In the field of few-shot learning, feature reweighting for dual-branch object detection is a popular solution (Köhler et al., 2021). In dual-branch few-shot object detection, the channel reweighting of the support set vector to the query set vector plays a key role in few-shot learning. Following this idea, we aim to build a reweighting module in our single backbone to help our model quickly adapt to new samples. However, since the backbone has yet to see this new category, it cannot accurately extract information. Therefore, we design a channel-level feature rectification module that can replace the feature interaction stage in two-stage few-shot training. We insert it into the backbone so the backbone can perform channel correction on the generated feature vector according to the image domain information in the new domain during the training process.

In the backbone network, a common view is that we can extract the different scales of features from different layers (Lin et al., 2017a; Li et al., 2022d). Inspired by this point, we uniformly insert the FCM into the backbone network to address the instance-level and image-level domain shifts.

The Feature Correction Module (FCM) we designed is shown in the lower part of Figure 5, and then we insert it into CSPDarknet53, which is the backbone of YOLOv5, as shown in the upper part of Figure 5. In each FCM, there are two branches. The first branch saves the raw input feature maps, and the second generates a reweighting vector to correct the feature maps of the first branch. Suppose the input feature map size is h*w*c. In the second branch, the feature map will first go through a global avg-pooling operation to obtain a 1*1*c vector followed by c groups depthwise convolution. Next, we feed the output 1*1*c vector to a sigmoid activation layer to normalize and reweight vectors. At last, the reweighted vectors will multiply the feature maps of the first branch to obtain the final outputs

4 Experiment

In this section, we will introduce the experimental results of our method and other methods in different scenarios. The experimental results are represented by the mean average precision (mAP) with an IOU threshold of 0.5. The mAP is determined by Precision and Recall. Precision represents the accuracy of the detected samples, and Recall represents the proportion of correctly detected samples among all correct samples.

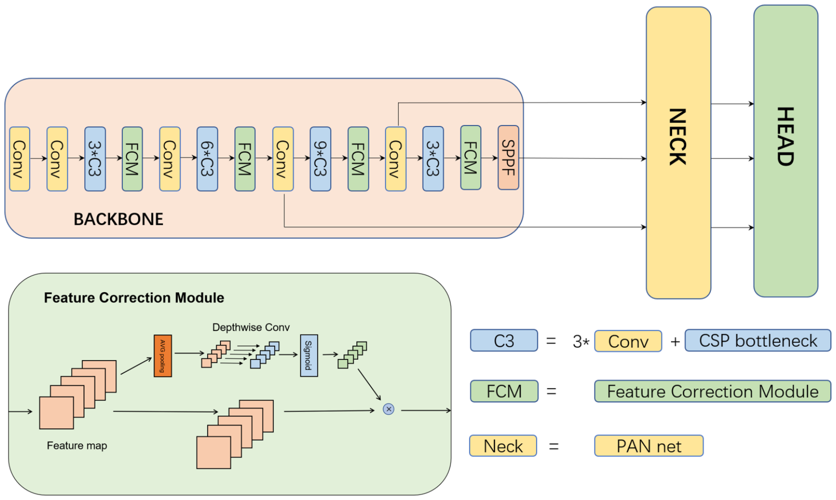

FIGURE 5
Our model structure and the specific implementation of the feature correction module.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Among them, TP refers to the true positive, which means the detected real samples; FN refers to the false negative, which means the correct samples that were not detected; and FP refers to the false positive, which means the falsely detected samples. The AP is determined by the area under the Precision-Recall (PR) curve, and we use the interpolation method to calculate it:

$$AP = \sum_{i=1}^{n-1}(Recall_{i+1} - Recall_i) \cdot Precision_{interp}(Recall_{i+1}) \tag{12}$$

Here mAP refers to the average value of each type of AP:

$$mAP = \sum_{i=1}^{k} AP_i \tag{13}$$

The mAP50 used in the following experiments means that the mAP score with an IOU threshold of 0.5. The adaptation time refers to the time required for each method to achieve the optimal effect in the target domain, and the time unit is hours (h).

4.1 Datasets

S-UODAC2020: This dataset was processed by Song et al. (Song et al., 2021). They used the style transfer model WCT2 (Yoo et al., 2019) to process the original UODAC2020 dataset into seven common underwater domains for evaluating domain adaptation, and each domain type has 791 images. type1-type6 is the source domain, and type7 is the target domain.

URPC2022[1]: URPC contains 9,000 images. The original dataset contains four categories, such as starfish. Here we only take the starfish category for analysis.

Aquarium[2]: The dataset consists of 638 images collected from two aquariums in the United States, which also contain the starfish class. Since the paired categories in the two data sets only include the starfish category, we use the starfish class from these two datasets (URPC and Aquarium) for cross-domain testing, in which URPC2022 is the source domain and Aquarium is the target domain.

4.2 Implementation details

Our code is based on official YOLOv5x(PyTorch)[3] with COCO dataset pre-training weights. Except for our proposed FCM, we do not adopt any other modules to modify the network. We adopt the Stochastic Gradient Descent (SGD) optimizer with a 0.01 learning rate and a 16 batch size. We set the picture size to 640 on the long side. All training time statistics are performed with a graphic card of GTX1080ti (11G).

4.3 Benchmark comparison

In Table 1, we compared two UDA methods including SCL (Shen et al., 2019) and SCAN (Li et al., 2022b) on the S-UODAC2020 dataset. The four columns (holothurian, echinus, scallop and starfish) in Table 1 represent the AP50 values of each category in the dataset, and mAP50 represents the average value of all categories. The time column represents the adaptation time of the algorithm when encountering a new domain, and the unit is hours. For the baseline, we used the network freeze strategy (freeze backbone) recommended by YOLOv5 to solve the few-shot problem (YOLOv5 w/ft). Since the dataset mainly includes image-level domain shifts, the number of targets in each picture is large, we adopt *shot = num* (X_t), and the performance results under ten shots are shown in Table 1. We can find that the UDA methods have poor accuracy under 10-shot. The two UDA methods also take a long time to adapt to each domain. Our method overcomes this problem with the only additional cost of labeling a few samples, which does not consume too much human effort.

In Table 2, we also compared methods such as SCL, SCAN, and SIGMA (Li et al., 2022c) on the URPC2022 and Aquarium dataset settings. The number of targets in the images of these datasets is relatively balanced, and there are image-level and instance-level domain offsets at the same time, so we strictly use the method to count. We provide the results under 3-shot and 10-shot in Table 2. The experimental results of “YOLOv5 w/ft” shown in Table 2 freezing the backbone module and fine-tuning the header module are better than “YOLOv5 w/o ft” but worse than our results. That means freezing the backbone module and fine-tuning the header module (YOLOv5 w/ft) can correct the domain shift to a certain extent, but the efficiency is lower than ours (freezing the head module and updating the backbone).

It can be seen that the classic UDA methods (SCL, SCAN, and SIGMA) cannot work with a small number of samples, and their time to adapt to the unfamiliar domain is much longer than our method, so they cannot quickly adapt to the unfamiliar domain.

4.4 Ablation analysis

To validate each component of our method, we design an ablation study on the S-UODAC dataset, as shown in Table 3. The “bb-ft” represents our migration learning strategy, and the “FCM” denotes the feature correction module. The four columns before the mAP column in Table 3 represent the AP50 values of each category in the dataset, and mAP50 represents the average value of all categories. Both the feature correction module and the migration learning strategy can significantly improve the

[1] http://www.urpc.org.cn/

[2] https://universe.roboflow.com/data-science-day-dry-run/aquarium-6cfzm/dataset/1.

[3] https://github.com/ultralytics/yolov5

TABLE 1 Our comparison results with other methods on the S-UODAC dataset.

method	holothurian	echinus	scallop	starfish	mAP50	Time
SCL	0.491	0.725	0.589	0.345	0.546	13.2h
SCAN	0.399	0.745	0.469	0.252	0.466	6.9h
YOLOv5 w/ft	0.604	0.780	0.707	0.587	0.669	0.19h
Ours	0.613	0.804	0.722	0.685	0.706	0.19h

TABLE 2 Our comparison results with other methods on the URPC2022 and Aquarium dataset.

method	3-shot		10-shot	
	mAP50	Time	mAP50	Time
SCL	0.349	14.3h	0.478	15.6h
SCAN	0.545	5.1h	0.607	6.2h
SIGMA	0.636	6.6h	0.652	6.5h
YOLOv5 w/o ft	0.516	–	0.516	–
YOLOv5 w/ft	0.685	0.1h	0.714	0.14h
Ours	0.710	0.09h	0.736	0.11h

TABLE 3 Our ablation experiments on the S-UODAC dataset.

method	bb-ft	FCM	holothurian	echinus	scallop	starfish	mAP50
Benchmark			0.425	0.803	0.647	0.519	0.599
Ours	✓		0.621	0.783	0.703	0.617	0.681
		✓	0.580	0.798	0.725	0.608	0.678
	✓	✓	0.613	0.804	0.722	0.685	0.706

✓ represents the training method using the column.

performance of the baseline model. We achieve the best result when these two components work simultaneously.

In Table 4, we tested with the activation functions in FCM and found that the sigmoid function performs slightly better than rectified linear unit (ReLU) in performance. For the three challenging categories, the sigmoid function leads to significant improvements. We conclude that this is because the sigmoid normalizes the vector between 0 and 1, which helps the final reweighting of our feature correction module. We also found that the FCM module with the sigmoid function converges faster than the case with the ReLU function. The result also verifies the point of attention mechanisms in recent years (Vaswani et al., 2017).

We visualize the results of the ablation experiments. The green bounding box in the figure refers to the correct sample missed by the detector. Figure 6A results from the benchmark training only on the source domain. The model missed many instances when we performed a cross-domain test. The results in Figure 6A also show the shortcomings of current detectors in cross-domain detection performance. Figure 6B shows the two-stage training method's result. We can see that the fine-tuning process can significantly reduce the number of missed samples, but some samples are still undetected. Figure 6C is the result of using the two-stage training method and FCM at the same time. It can be seen that our method has only one missed target and no false detections. Based on the result in Figure 6, we can conclude that both the proposed two-stage training method and FCM can efficiently resist the performance degradation from the cross-domain detection task.

To further verify the attention improvement, Figure 7 shows some examples using the Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017) image under different datasets. Gradient-weighted Class Activation Mapping can reflect which part of the image the neural network pays

TABLE 4 Performance of different activation functions on the S-UODAC dataset.

activation	holothurian	echinus	scallop	starfish	mAP50
ReLU	0.596	0.814	0.697	0.683	0.698
Sigmoid	0.613	0.804	0.722	0.685	0.706

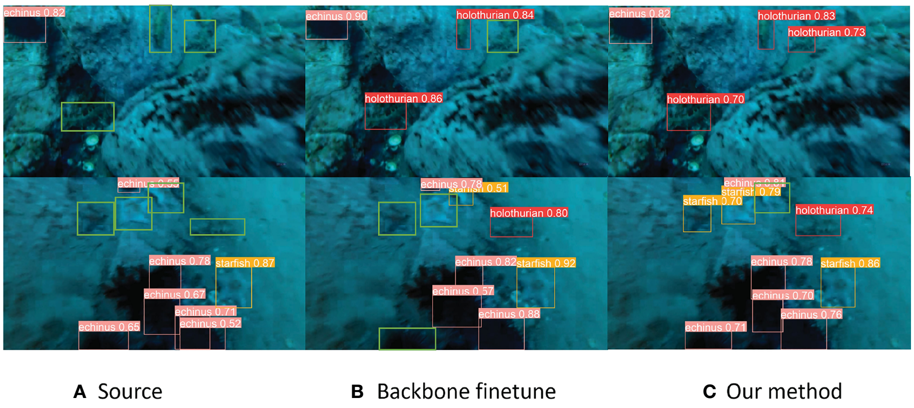

FIGURE 6
Visualization results of ablation experiments.

attention to when detecting and recognizing a certain type of object. The redder the color of the heat map, the more the network pays attention to this part. Figure 7A contains three raw images; Figure 7B shows the results of YOLOv5 trained only on the source domain. We can see that many target areas are inactivated during the detection process. In other words, the network has not paid attention to these areas. Figure 7C represents the Grad-CAM results of our method. All target regions are accurately activated after fine tuning with our approach. The heat map visualization results indicate that our method can better locate the object in the new domain. The heat map visualization results can also prove the above point of view. Figure 7B (freezing the backbone module and fine-tuning the header module) performs worse than Figure 7C (freezing the header module and fine-tuning the backbone module). Our network paid attention to these targets without missing the original detected samples, indicating that the extracted features are offset from the actual feature space when the backbone is not adapted to the target domain.

When we select the final weight, we adopt the “early stop” strategy, which allows us to obtain the training weight when the loss of the verification set is the smallest. Figure 8 is the loss curve image during our fine-tuning process. In the “early stop” strategy, a commonly used parameter is “patience”. Assuming its value is n, it means that if the result of the k_{th} epoch training is still the best

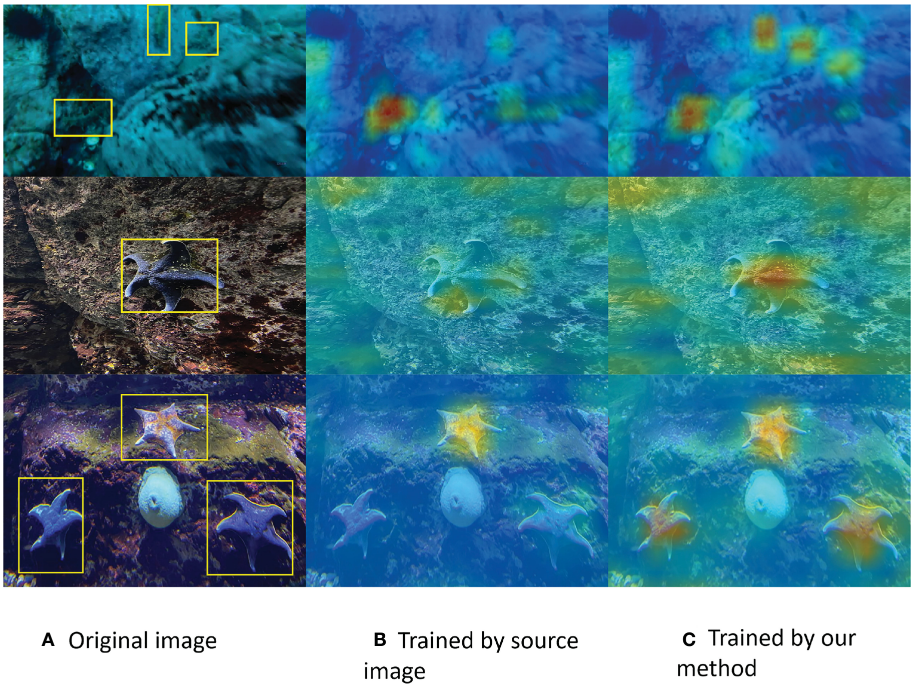

FIGURE 7
Our Grad-CAM images under different datasets.

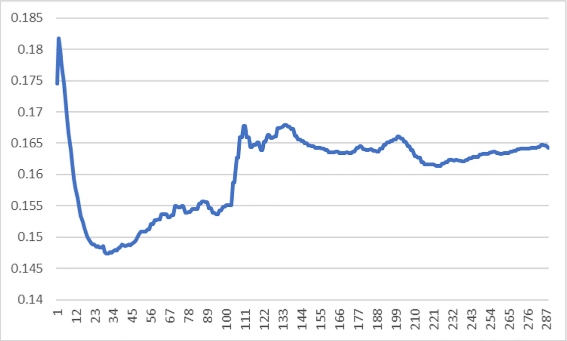

FIGURE 8
Our loss curve chart.

after n epochs, stop the training. The weight of the k_{th} epoch is selected as the final weight. We set n to 250. From Figure 8, we can find that our method quickly converged in about 30_th epoch. Then the curve gradually grew up. Since the lowest point is clear, a large enough can easily locate the lowest point.

We also test our model in a short underwater video to prove the superiority of our method for object detection in unfamiliar waters. Figure 9 shows the detection result of one frame. The left side is our method, and the right is the fine-tuning results after pre-training on a large-scale dataset (COCO) of YOLOv5. Our approach is significantly ahead of the comparison method in both recall and precision, and our fine-tuning uses the first frame of the video. More details can be found in our GitHub project.

5 Discussion

Currently, deep learning has achieved remarkable results in computer vision and has also produced good results in underwater computer vision, such as underwater observation and underwater image processing. However, its data-driven models also have limitations. As discussed in the article, deep learning models have shown a significant performance drop in test scenarios in an unfamiliar environment with different data distributions from the training set. Previous works Chen et al. (2018); Ganin et al. (2016) have shown that the main reason for cross-domain performance degradation in tasks such as classification and object detection is that the backbone cannot extract domain-invariant features.

In the field of underwater vision, we have an urgent need for domain adaptation algorithms:

- Underwater images are affected by plankton and river flooding disasters, often resulting in large changes in image colors.
- In different water domains, due to environmental influences, biological morphology often has certain changes.
- Many different species of the same family have certain differences in appearance, which also brings about domain shifts.

Regardless of the data domain in which the target category appears, humans can accurately capture the invariant features in different domains to complete classification and labeling. Inspired by this point, many researchers trained the backbone through domain adversarial training and other strategies, which can make the backbone extract domain invariant features. However, this training method requires a large number of target domain samples, which is very difficult to obtain in underwater scenarios. Unfortunately, we often need more training samples to adapt to the test scenario, especially when underwater data collection is challenging.

We propose a few-shot domain adaptation object detection algorithm based on a two-stage training strategy and an FCM module, which can quickly adapt to the target domain with only a small number of annotated samples, not only solving the defects of previous domain adaptation work under few-shot but also being more suitable for underwater scene applications. However, our method still has some drawbacks. When the algorithm adapts to the target domain, it does not consider catastrophic forgetting. Because we only use target domain samples to fine-tune the network rather than jointly training with source domain samples, this inevitably leads to a performance drop in the source domain.

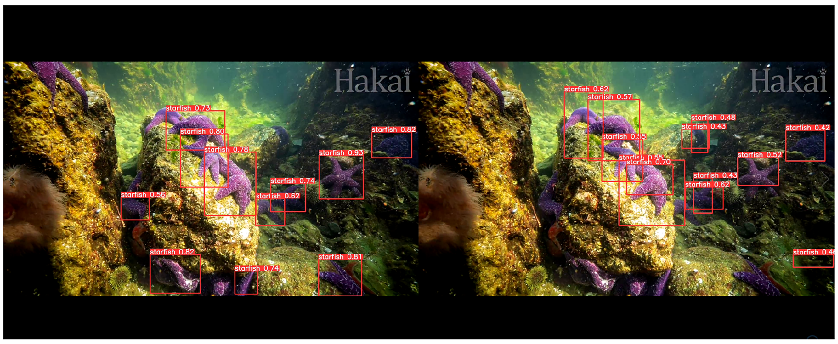

FIGURE 9
Demo on a YouTube video, the confidence threshold is 0.4 and the IOU threshold is 0.45.

Our current solution to this problem is to retain weight files for each domain so that they can be used at any time.

6 Conclusion

This paper proposes a novel few-shot domain adaptive object detection framework. Our algorithm can transfer the object knowledge information from the source domain to the target domain, achieving a situation where only a small number of annotated target domain samples are used. At the same time, our algorithm also inspires unsupervised few-shot domain adaptive object detection, such as exploring the use of an image-to-image translation model to generate a small number of target domain samples for training using our method.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/roadhan/FSCW.

Author contributions

LH completed most of the work in this paper. JZ completed the synthesis of the datasets and the typesetting of the paper. ZY handled the work of revising the article, and BZ provided guidance and funding for this research. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62171419), the Project of Sanya Yazhou Bay Science and Technology City (Grant No. SCKJ-JYRC-2022-102) and Hainan Province Science and Technology Special Fund, China (ZDYF2022SHFZ318).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Cai, L., Zhang, Z., Zhu, Y., Zhang, L., Li, M., and Xue, X. (2022). "BigDetection: a large-scale benchmark for improved object detector pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 4777–4787.

Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. (2018). "Domain adaptive faster r-CNN for object detection in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3339–3348.

Fan, B., Chen, W., Cong, Y., and Tian, J. (2020a). "Dual refinement underwater object detection network," in *Proceedings of the European Conference on Computer Vision*. 275–291.

Fan, Q., Zhuo, W., Tang, C.-K., and Tai, Y.-W. (2020b). "Few-shot object detection with attention-RPN and multi-relation detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4013–4022.

Fu, C., Liu, R., Fan, X., Chen, P., Fu, H., Yuan, W., et al. (2023). Rethinking general underwater object detection: datasets, challenges, and solutions. *Neurocomputing* 517, 243–256. doi: 10.1016/j.neucom.2022.10.039

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* 17, 2096–2030.doi: 10.48550/arXiv.1505.07818

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., and Darrell, T. (2019). "Few-shot object detection *via* feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8420–8429.

Kiran, M., Pedersoli, M., Dolz, J., Blais-Morin, L.-A., Granger, E., et al. (2022). Incremental multi-target domain adaptation for object detection with efficient domain transfer. *Pattern Recognition* 129, 108771. doi: 10.1016/j.patcog.2022.108771

Köhler, M., Eisenbach, M., and Gross, H.-M. (2021). Few-shot object detection: a comprehensive survey. *arXiv preprint arXiv* 2112, 11699. doi: 10.1109/TNNLS.2023.3265051

Lee, H., Lee, M., and Kwak, N. (2022). "Few-shot object detection by attending to per-sample-prototype," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2445–2454.

Li, W.-H., Liu, X., and Bilen, H. (2022d). "Cross-domain few-shot learning with task-specific adapters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7161–7170.

Li, W., Liu, X., Yao, X., and Yuan, Y. (2022b). "SCAN: cross domain object detection with semantic conditioned adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 6. 7.

Li, W., Liu, X., and Yuan, Y. (2022c). "SIGMA: semantic-complete graph matching for domain adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5291–5300.

Li, B., Wang, C., Reddy, P., Kim, S., and Scherer, S. (2022a). "AirDet: few-shot detection without fine-tuning for autonomous exploration," in *Proceedings of the European Conference on Computer Vision*. 427–444.

Liang, X., and Song, P. (2022). "Excavating roi attention for underwater object detection," in *Proceedings of the IEEE International Conference on Image Processing*. 2651–2655.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft COCO: common objects in context," in *Proceedings of the European Conference on Computer Vision*. 740–755.

Lin, W.-H., Zhong, J.-X., Liu, S., Li, T., and Li, G. (2020). "ROIMIX: proposal-fusion among multiple images for underwater object detection," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. 2588–2592.

Liu, H., Song, P., and Ding, R. (2020). WQT and DG-YOLO: towards domain generalization in underwater object detection. *arXiv preprint arXiv* 2004, 06333. doi: 10.48550/arXiv.2004.06333

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 779–788.

Saito, K., Ushiku, Y., Harada, T., and Saenko, K. (2019). "Strong-weak distribution alignment for adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6956–6965.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-CAM: visual explanations from deep networks *via* gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.

Shen, Z., Maheshwari, H., Yao, W., and Savvides, M. (2019). SCL: towards accurate domain adaptive object detection *via* gradient detach based stacked complementary losses. *arXiv preprint arXiv* 1911, 02559. doi: 10.48550/arXiv.1911.02559

Song, P., Dai, L., Yuan, P., Liu, H., and Ding, R. (2021). Achieving domain generalization in underwater object detection by image stylization and domain mixup. *arXiv preprint arXiv* 2104, 02230.

Sun, B., Li, B., Cai, S., Yuan, Y., and Zhang, C. (2021). "FSCE: few-shot object detection *via* contrastive proposal encoding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7352–7362.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30:6000–6010. doi: 10.5555/3295222.329534

Wang, X., Huang, T. E., Darrell, T., Gonzalez, J. E., and Yu, F. (2020). Frustratingly simple few-shot object detection. *arXiv preprint arXiv* 2003, 06957. doi: 10.48550/arXiv.2003.06957

Wang, T., Zhang, X., Yuan, L., and Feng, J. (2019). "Few-shot adaptive faster r-CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7173–7182.

Yoo, J., Uh, Y., Chun, S., Kang, B., and Ha, J.-W. (2019). "Photorealistic style transfer *via* wavelet transforms," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9036–9045.

Yu, F., Wang, D., Chen, Y., Karianakis, N., Shen, T., Yu, P., et al. (2022). "SC-UDA: style and content gaps aware unsupervised domain adaptation for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 382–391.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). "Distance-IoU loss: faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*. 12993–13000.

Zhu, C., Chen, F., Ahmed, U., Shen, Z., and Savvides, M. (2021a). "Semantic relation reasoning for shot-stable few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8782–8791.

Zhu, X., Lyu, S., Wang, X., and Zhao, Q. (2021b). "TPH-YOLOv5: improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2778–2788.