IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 21, 2020, accepted July 30, 2020, date of publication August 7, 2020, date of current version August 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014910

Small-Object Detection in UAV-Captured Images via Multi-Branch Parallel Feature Pyramid Networks

YINGJIE LIU, FENGBAO YANG, AND PENG HU
School of Information and Communication Engineering, North University of China, Taiyuan 030051, China

Corresponding author: Fengbao Yang (yfengb@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972363, in part by the Key Projects of Key Research and Development Plan in Shanxi Province under Grant 201903D421043, and in part by the Graduate Education Innovation Project of Shanxi province, China, under Grant 2020SY365.

ABSTRACT Small object is one of the primary challenges in the field of object detection, which is notably pronounced to the detection in the images from Unmanned Aerial Vehicles (UAV). Existing detectors based on deep-learning methods usually apply the feature extraction networks with a large down-sampling factor to obtain higher-level features. However, such big stride tends to make the feature information of small objects become the little point or even vanish in the low-resolution feature maps due to the limitation of pixels. Therefore, a novel structure called Multi-branch Parallel Feature Pyramid Networks (MPFPN) is proposed in this article, which aims to extract more abundant feature information of the objects with a small size. Specifically, the parallel branch is designed to recover the features that missed in the deeper layers. Meanwhile, a supervised spatial attention module (SSAM) is applied to weaken the impact of background noise inference and focus object information. Furthermore, we adopt cascade architecture in the Fast R-CNN stage for a more powerful localization capability. Experiments on the public drone-based datasets named VisDrone-DET demonstrate that our method achieves competitive performance compared with other state-of-the-art detection algorithms.

INDEX TERMS Unmanned aerial vehicle, object detection, multi-branch parallel feature pyramid networks (MPFPN), feature fusion, cascade architecture.

I. INTRODUCTION

UAVs have initially appeared in the last 1920s and are mainly used in the military due to the advantages, i.e., small volume, convenient operation, and strong survival ability on the battlefield [1]. In recent years, with the gradual advance of UAV technology, the civil UAV has been widely applied in the world of real-life, including agricultural production [2], aerial photography [3], fast delivery [4], environmental monitoring, etc.

Simultaneously, with the rapid progress of deep convolutional neural networks (DCNNs), deep-learning-based methods [5]–[8] have achieved a significant breakthrough in various kinds of the field in computer vision. As for object detection, many detection methods have been proposed and reached great success in natural image detection(e.g., images in Pascal VOC [9], MS COCO [10]), such as R-CNN series [11]–[13], YOLO series [14]–[17], SSD series [18], [19]. However, there exist many apparent differences between natural images and aerial images. For example, the flight altitude of civil UAV is usually tens to hundreds of meters which bring about a large scene to UAV-captured images. In this situation, most objects are generally small size and different viewpoints, which make object detection in aerial images keep a hard challenge [20]. Figure 1 shows the objects and corresponding annotated boxes. It can be easily seen from the picture that the objects are not only small size but also large numbers. This problem has become one of the main factors that weaken the performance of current popular detectors.

Recently, many methods [21]–[26] have been proposed and try to solve the issue of small objects. However, in the UAV-captured image, the object detection based on deep learning still faces severe challenges. For example, Faster R-CNN only uses the last layer (i.e., high-level features) to implement the prediction, which does not adequately consider the features of other layers, resulting in the apparent shortage of its detection ability in the detection of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhua Guo.

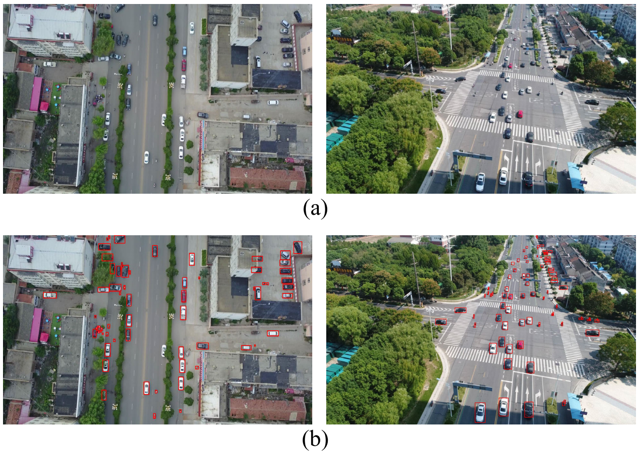


FIGURE 1. Visualization of UAV-captured images. (a) Objects with a small size and different viewpoint; (b) Ground-truth.

small targets. To solve this issue, SSD has been proposed which extracts the features from different layers to predict.

However, it predicts from the deeper layer and still does not consider the characteristics of the shallow layer. Therefore, its detection ability for small targets is still not good. Later, the classical feature fusion method is proposed, which combines the multiscale feature information from the different layers. For instance, the Feature Pyramid Network (FPN) [27] fuses the low-level features and the high-level features by adopting top-down architecture and lateral connections, which significantly improve the performance when detecting the objects with small size. However, to obtain a larger receptive field and higher-level features, most feature extractors [28]–[30] has a large stride, such as 32× strides in ResNets [29]. Such a large stride will directly make the small objects become a little point or even disappeared in the deeper layers. Notably, it makes the later up-sampling operation cannot recover its missed feature information efficiently. This problem imposes restrictions on the representation capability of FPN, in other words, the feature information carried by its pyramid layers is still insufficient to detect small objects effectively.

In this article, in order to extract more contextual semantic information of small objects, we propose a novel structure called Multi-branch Parallel Feature Pyramid Networks (MPFPN), which aims to obtain sufficient feature information of small objects. In the proposed MPFPN, we first design two additional parallel branches to start up-sampling operation from the shallower layers for recovering more feature information that missed in the deeper layers. Meanwhile, to deal with the background noise, a supervised spatial attention module (SSAM) is added in the lateral connections of MPFPN. Finally, all pyramid layers, including one initial top-down pathway and two parallel pathways, are fused for the final prediction. Besides, to obtain a more powerful localization capability for objects in aerial images, cascade architecture with three stages is adopted to refine the bounding box prediction in the Fast R-CNN stage. In summary, the main contributions of this article are listed as follows.

1. A novel and effective structure called Multi-branch Parallel Feature Pyramid Networks (MPFPN) is firstly proposed for the object detection in UAV aerial images.

2. the Supervised Spatial Attention Module (SSAM) is designed to against complex background noise and highlight the foreground information with a supervised learning method in MPFPN.

3. Cascade architecture is applied in the Fast R-CNN stage to refine the bounding box regression and enhance the capability of locating the objects.

4. Extensive experiments demonstrate the state-of-the-art performance on the drone-based image dataset VisDrone-DET.

The rest of this article is organized as follows. Section 2 gives the related work of general object detection and small object detection. Section 3 shows the analysis and description of the proposed framework. Section 4 presents experiments of the drone-based dataset. Discussions of the experiment result are discussed in Section 5. The last section concludes this article.

II. RELATED WORKS

Small-object detection keeps an extremely challenging for a long term in the field of computer vision. In the past few years, Convolutional Neural Networks (CNNs) have made a breakthrough development for object detection. In general, the current detection methods can be mainly divided into two categories, the one is one-stage detectors, and the other is two-stage detectors.

Generally, the one-stage detectors are highly efficient due to a direct prediction of predefined anchors without the process of proposal generalization. In YOLO [14], it directly predicts the bounding box's coordinates from an image that is gridded into several regions. YOLO9000 [15] and YOLO-v3 [16] take the experience from other works to achieve higher accuracy, such as applying prior anchor mechanism and feature fusion network. SSD [18] sets the anchor box on the feature maps with different scale and then directly make the anchor classify and regress.

Although the one-stage detectors have a high speed, its detected accuracy are generally lower than the two-stage detectors. The reason is that the two-stage detectors generate and choose the suit region proposal at first and then refine them in the next stage. Thus, they usually have a better performance on accuracy. Faster R-CNN [13] improves the Fast R-CNN [12] and R-CNN [11] by designing region proposal network (RPN) to generate the proposals, which not only improves the speed and accuracy but also allows the detection networks to implement end-to-end training. R-FCN [31] further improves the accuracy by adopting fully convolutional neural networks. However, the above detectors perform badly on the small objects due the inefficient utilization of low-level features and high-level features. Therefore, these detectors that simply utilize the features of a single layer usually reach a lower accuracy in the image that has large number of small objects, such as UAV-captured images.

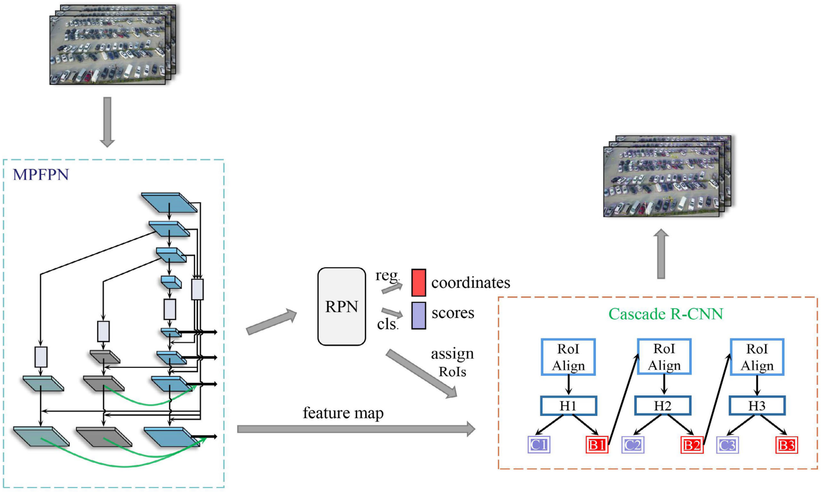


FIGURE 2. The architecture of entire framework.

In UAV-captured images, the proportion of small objects is very large in most cases. Generally, these small objects pose a massive challenge to the recognition due to the small number of pixels. To solve the problem of small objects, FPN [27] achieves significant progress by adopting feature fusion networks. As shown in Figure 3., FPN includes three main parts, namely the bottom-up architecture, top-down architecture, and lateral connection that fuse the multi-scale features. In order to overcome the problem that the small objects vanish in the deeper layers. DetNet [32] attempt to maintain a high-resolution feature map by applying the reduced layers of ResNet architecture to avoid the disappearance of small objects, which aims to upsample the feature information of small objects from the shallow layers, furthermore, to compensate the discarded high-level features, it applies dilated convolution to expand the receptive field. The proposed MPFPN is inspired by the architecture of FPN and DetNet, which can not only retain the high-level features of deeper layers but also recover sufficient feature information of small objects from the shallow layers.

Besides, the complicated scene usually brings an intense background noise and result in a bad performance for UAV-captured images. Attention mechanism [33]–[35] has proved that it could effectively deal with the background noise interference issue. The primary purpose of attention mechanism is to focus on the objects of interests and weaken the background inference existing in the images. In [34], it proposed the multi-dimensional attention mechanism to weaken the background noise and highlight the foreground information, including the guided spatial attention and channel attention. Recently, researchers [36]–[39] also make progress on the regression process for the general object detection. Cascade R-CNN [39] adopts the multi-stage detection sub-network with different IoU threshold to refine the bounding box regression. In this article, the proposed framework gathers the feature fusion network, attention module and the cascade architecture into the object detection in UAV-captured images. The relevant details will be presented in section 3.

III. THE PROPOSED METHOD

The framework of the proposed method is shown in Figure 2, which mainly includes two parts, i.e., MPFPN for the first stage and Cascade architecture for the Fast R-CNN stage. Specifically, MPFPN aims to extract more powerful and sufficient feature information from feature maps and fuse these features before the final prediction of each pyramid layer. Then the proposals are generated from the region proposal networks (RPN) for the later stage. Finally, high-quality regression and classification of proposals are processed by cascade architecture, namely Cascade R-CNN.

A. MULTI-BRANCH PARALLEL FEATURE PYRAMID NETWORKS

In order to obtain higher-level features, previous works usually adopt a large factor of down-sampling, which does well in the classification task. For example, ResNets is a classical kind of classification network with a very deep layer, and the spatial resolution of its feature map is 32× sub-sampled, which obtains the larger receptive field and higher-level features. This is beneficial to identify the objects with large size. However, 32× down-sampling is quite unfavorable to detect the objects with a small size because these objects will be easily vanished in the deep layer because of a limited pixel. So previous works that only use a single large stride network have a poor performance on small objects, and such weakness is quite obvious to the objects in the aerial images. Feature Pyramid Networks (FPN) adopts top-down architecture and lateral connections to weaken the impact from the small objects. However, since the small objects have vanished in the deeper layers, its contextual semantic information will disappear simultaneously, and the following up-sampling operation cannot effectively recover it. Therefore, there is still much feature information that missed in feature maps that can be exploited.

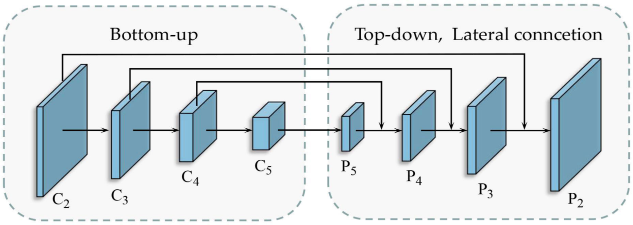


FIGURE 3. The architecture of feature pyramid network.

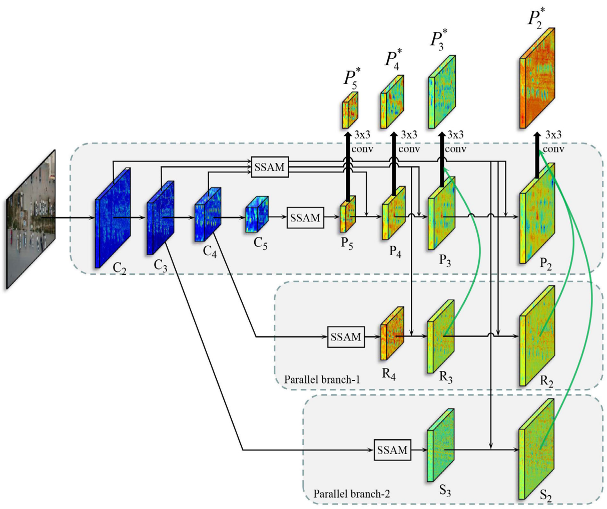


FIGURE 4. The architecture of multi-branch parallel feature pyramid networks.

In this work, a novel network called MPFPN is proposed for the first time, which increases two additional parallel branches to obtain more contextual semantic features of small objects. Figure 4 gives the details of MPFPN, and the detailed description is as follows.

1) FEATURE PYRAMID NETWORKS (FPN)

We describe the last feature map from each residual block of ResNets as $\{C_2, C_3, C_4, C_5\}$ (see above Figure 3) and the strides of its corresponding layers are {4, 8, 16, 32} pixels. In top-down architecture, the feature maps of pyramid layers are described as $\{P_2, P_3, P_4, P_5\}$.

2) MULTI-BRANCH PARALLEL ARCHITECTURE

As shown in Figure 4, we increase two additional parallel branches, which operates up-sampling and lateral connection from the corresponding layers. Then these pyramid layers are combined to gather its features. Here the first parallel branch is described as the feature maps $\{R_2, R_3, R_4\}$, and the second parallel branch is defined as $\{S_2, S_3\}$. The specific definition is as follows:

$$\begin{aligned} P_5^* &= P_5, P_4^* = P_4 \\ P_3^* &= Conv_{3\times3}\{P_3 \oplus R_3\} \\ P_2^* &= Conv_{3\times3}\{P_2 \oplus R_2 \oplus S_2\} \qquad (1) \end{aligned}$$

where $Conv_{k\times k}$ indicates the k×k convolution, k is the kernel size. $\oplus$ is the operation of concatenation. P_i^* represents the prediction of all fused feature maps from parallel branches R_i or S_i. Finally, we get the final prediction as to the feature maps $\{P_2^*, P_3^*, P_4^*, P_5^*\}$. Besides, each feature map of pyramid layer is defined as:

$$\begin{aligned} P5 &= Conv_{1\times1}(C5) \\ P_i &= Conv_{3\times3}\left[Conv_{1\times1}(C_i) + Upsample(P_{i+1})\right] \qquad (2) \end{aligned}$$

where $Conv_{1\times1}$ is applied to make the number of the channel to 256. *Upsample* represents bilinear up-sampling in this article. Meanwhile, the parallel branch is similar to the definition of the pyramid layer, for example, feature maps of the parallel branch-1 is defined as:

$$\begin{aligned} R_4 &= Conv_{1\times1}(C_4) \\ R_i &= Conv_{3\times3}\left[Conv_{1\times1}(C_i) + Upsample(R_{i+1})\right] \qquad (3) \end{aligned}$$

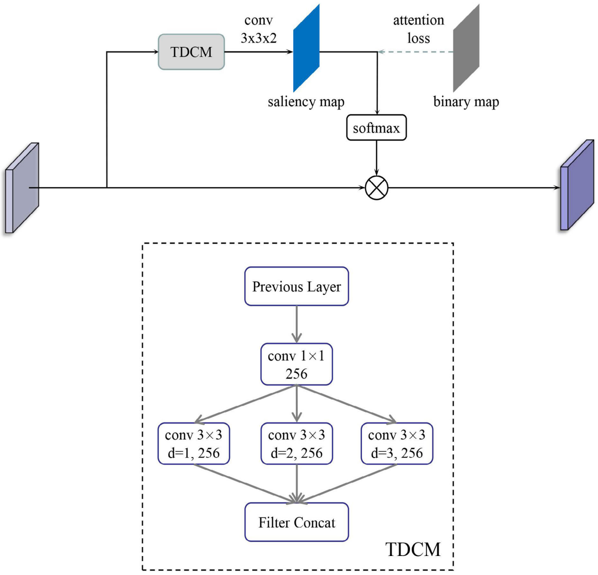
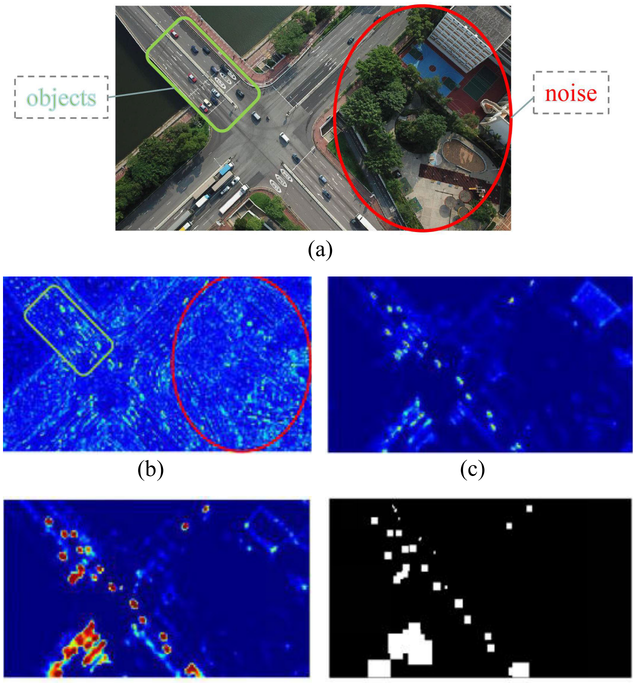


FIGURE 5. The architecture of supervised spatial attention module.

3) SUPERVISED SPATIAL ATTENTION MODULE (SSAM)

In UAV-captured images, the complexity of the scene usually brings strong noise interference. As shown in Figure 6a, excessive noise such as buildings or trees can disturb small objects' recognition. Therefore, inspired by [34] and [40], the supervised spatial attention module called SSAM is designed into MPFPN to weaken the impact of the background noise and highlight the foreground information. Figure 5 gives its detailed architecture, as shown in the picture, the input feature map first goes through the Trident Dilated Convolution Module (TDCM), which contains three different rates of dilated convolution to capture the multi-scale objects in aerial images. Then, the saliency map (see figure 6d) is obtained by the convolution operation with 2-channel output. Meanwhile, we get the binary map (see figure 6e) from the ground-truth label, and then a supervised learning method is utilized to guide the saliency map to learn its instance mask. Next, softmax operation is performed, and then one of its channels is selected to multiply with



(a)

(b)

(c)

(d)

(e)

FIGURE 6. The visualization of the supervised spatial attention module. (a) Input image; (b) feature map before applying SSAM; (c) feature map after applying SSAM; (d) Saliency map; (e) Binary map, the color black is the background, the color white is the instance mask of the ground truth.

the input feature map. Finally, the new output feature map is obtained after the above process. Note that we adopt the cross-entropy loss as the attention loss function.

As shown in Figure 6b, there is much background noise in the feature map, which can overwhelm the information of small objects. Figure 6c is the visualization result after being processed by SSAM, and it can be seen that the background noise is weakening obviously, and the object information is also enhanced in the meantime. Besides, we can observe a clear boundary of objects in Figure 6c, which helps avoid missed detection.

B. CASCADE ARCHITECTURE

In the first stage of the proposed framework, we have introduced MPFPN to improve the capability of extracting feature information for small objects. However, in the Fast R-CNN stage (i.e., the second stage), most works pay less attention to the precise localization for objects in aerial images, which usually bring a bad accuracy when detecting with the high IoU (intersection over union) threshold. Therefore, we replace the initial single regressor with cascade architecture [39] in the Fast R-CNN stage to enhance the localization capability for the objects in UAV-captured images. Furthermore, we replace the RoI pooling with RoI Align [41] to avoid the feature misalignment. As is depicted in Figure 7, the architecture refines the bounding box regression by utilizing a multi-stage detection sub-network, $\{H_1, H_2, H_3\}$ represents the head of each network, $\{B_1, B_2, B_3\}$ indicates the coordinates of the bounding box, $\{C_1, C_2, C_3\}$ is the classification score.

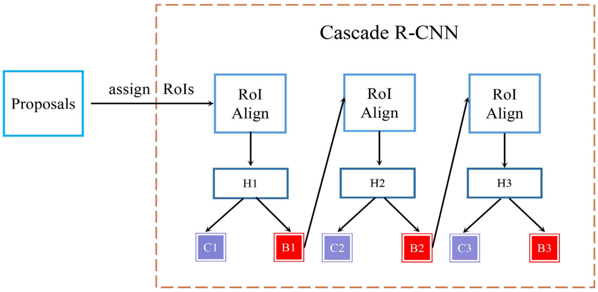


FIGURE 7. The architecture of Cascade R-CNN.

Bounding box regression in the cascade architecture is defined as:

$$f(x, \mathbf{b}) = fT \circ fT - 1 \circ \cdot \cdot f1\,(x, \mathbf{b}) \tag{4}$$

where the bounding box $\mathbf{b} = \left(b_x, b_y, b_w, b_h\right)$ contains four coordinate values of an image path x, T represents the number of the total cascaded stages. Due to an appropriate value of IoU threshold is used to each corresponding stage, all of the cascaded regressors $\{fT \circ fT - 1 \circ \cdot \cdot f1\,(x, \mathbf{b})\}$ can be optimized with the sample distribution $\left\{\mathbf{b}^t, \mathbf{b}^{t-1} \cdots \mathbf{b}^1\right\}$. Therefore, cascade architecture can improve the operation of the bounding box regression effectively.

Taking into account the characteristics of aerial images, we set the hyper-parameter T to 3 in all experiments, which means three cascaded stages are adopted. Meanwhile, to match the size of most objects, we assign the IoU threshold of positive RoIs to each stage to {0.5, 0.6, 0.7}, specifically, 0.5 is applied for the first stage, 0.6 for the second stage, 0.7 for the third stage. These settings ensure a high-quality regression for the objects with a small size. Figure 8 gives the visualization of negative RoIs and positive RoIs in three stages with different IoU thresholds, it can be seen a continuous refinement of bounding box regression.

C. LOSS FUNCTION

We use multi-task loss function in this work, which is defined as follow:

$$\begin{aligned} L = {} & \frac{1}{h \times w} \sum_{i}^{h} \sum_{j}^{w} L_{att}\left(u_{ij}, u_{ij}^{*}\right) \\ & + \frac{1}{N} \sum_{t=1}^{T} \sum_{n=1}^{N} L_{cls}\left(c_{tn}, c_{tn}^{*}\right) + \frac{\lambda}{N} \sum_{t=1}^{T} \sum_{n=1}^{N} L_{reg}\left(r_{tn}, r_{tn}^{*}\right) \end{aligned} \tag{5}$$

In this formula, L_{att} refers to the attention loss function, which uses softmax cross-entropy. h, w is the height and width of the labeled binary map. L_{cls} represents the classification loss calculated by softmax cross-entropy. loss function L_{reg} refers to the regression loss, which adopts the smooth L1 loss as defined in [12]. T represents the number of the total cascaded stage, and the hyper-parameter λ in Eq. 5 is set to balance different task, we set $T = 3$, $\lambda = 1$ in all experiments. Furthermore, the function L_{att}, L_{cls} and L_{reg} are

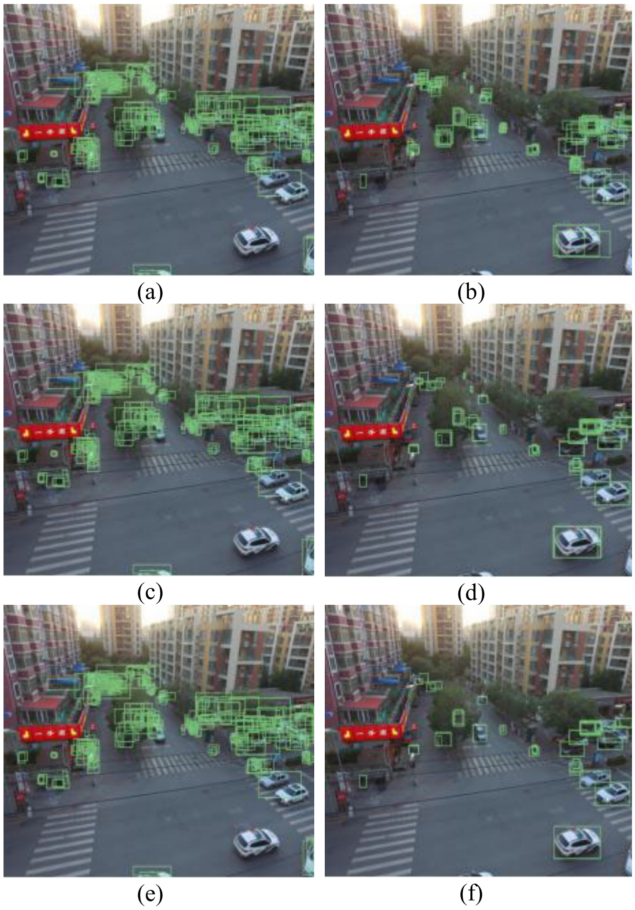


FIGURE 8. The visualization of negative RoIs (left-row) and positive RoIs (right-row) in three stages. (a), (b) the first stage; (c), (d) the second stage; (e), (f) the third stage.

respectively defined as follow:

$$L_{att}\left(u_{ij}, u_{ij}^{*}\right) = -u_{ij}^{*} \log u_{ij} \quad (6)$$

$$L_{cls}\left(c_{tn}, c_{tn}^{*}\right) = -c_{tn}^{*} \log c_{tn} \quad (7)$$

$$L_{reg}\left(r_{tn}, r_{tn}^{*}\right) = \text{smooth}_{L_1}\left(r_{tn} - r_{tn}^{*}\right) \quad (8)$$

$$\text{smooth}_{L_1}(x) = \left\{\begin{array}{l} 0.5x^2, |x| < 1 \\ |x| - 0.5, \textit{otherwise} \end{array}\right\} \quad (9)$$

IV. EXPERIMENTS AND RESULTS

In this section, the used datasets will be introduced at first, then we present four groups of experiments to explore the detection performance of the proposed framework. All experiments are constructed on GTX 1080Ti GPU and implemented in Python 3.6 using the Tensorflow framework with a version 1.12.

A. DATASETS AND EVALUATION METRICS

We demonstrate our framework on the drone-based dataset, namely VisDrone-DET [42], which has been released in http://aiskyeye.com/. The benchmark dataset focuses on four core problems, i.e., single-object tracking, object detection in images, crowd counting, and multi-object tracking. This article mainly aims to object detection in images, including 10,209 static images from drones. Ten common categories are involved, such as car, truck, pedestrian, bicycle, motor, etc.

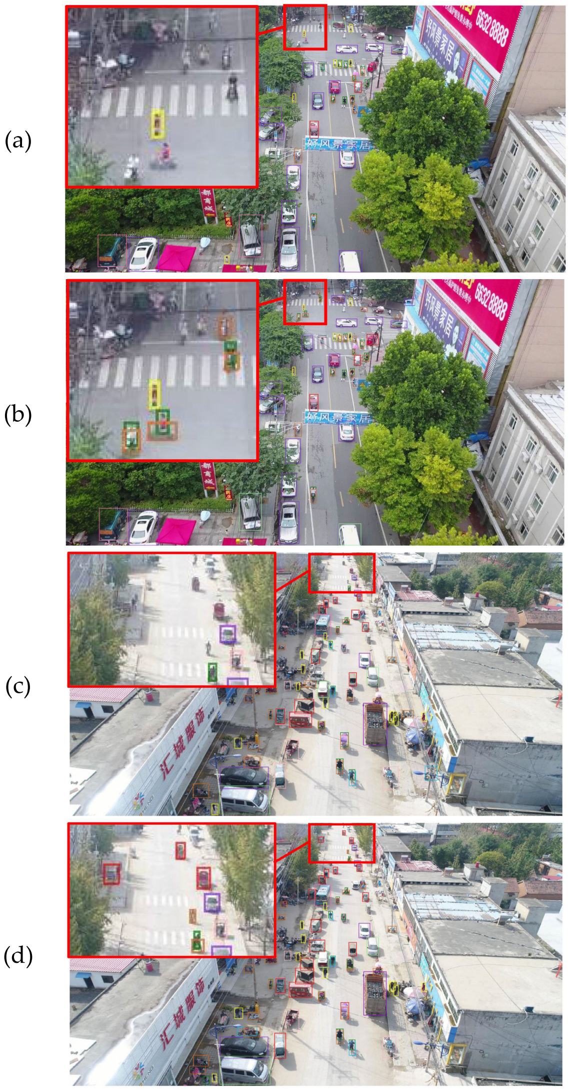


FIGURE 9. The visualization detection result of two methods on the VisDrone dataset. (a), (c) detection results of initial FPN; (b), (d) detection result of MPFPN. Different color represents its corresponding category.

Due to the used datasets has its evaluation method, in this article, its existed evaluation metrics evaluate the whole experiment results. According to the setting of dataset MS COCO [10], VisDrone uses the Average Precision (AP) and Average Recall (AR) metrics to measure the detection performance, including $AP_{0.5:0.05:0.95}$, $AP_{0.5}$, $AP_{0.75}$ and $AR_{500.}$ Specifically, $AP_{0.5:0.05:0.95}$ refers to the computation on the average value over all ten IoU thresholds from 0.5 to 0.95 with the step size 0.05. $AP_{0.5}$ and $AP_{0.75}$ are computed in a single IoU threshold 0.5 and 0.75 respectively. Also, the max detections per image are 500.

B. IMPLEMENTATION DETAILS

Extensive experiments in this article are measured on the validation set of VisDrone2020. We adopt the pre-training model ResNets-101 to initialize the network. Besides, Input

TABLE 1. Comparative results in each category. All categories are evaluated in AP$_{0.5:0.95}$. FPN: Feature Pyramid Networks. MPFPN: Multi-branch Parallel Feature Pyramid Networks. All methods use Cascaded R-CNN as the baseline network. Class ped. refers to the pedestrian, awn. refers to the awning-tricycle.

Baseline	Method	Test speed	AP [%]	ped.	person	bicycle	car	van	truck	tricycle	awn.	bus	motor
Cascade R-CNN	FPN	0.39s	27.0	25.36	17.99	9.92	54.66	34.45	26.81	17.45	11.43	44.2	23.85
	MPFPN	0.46s	29.05	26.38	18.73	12.41	57.85	36.92	28.25	21.14	12.01	45.42	26.49

TABLE 2. Comparative results in each category with five IoU threshold. All detection results are measured in the proposed framework.

Category	AP$_{0.5}$ [%]	AP$_{0.6}$ [%]	AP$_{0.7}$ [%]	AP$_{0.8}$ [%]	AP$_{0.9}$ [%]
car	87.88	83.82	73.61	52.81	13.45
van	56.49	54.02	47.44	33.37	7.0
truck	46.75	42.59	35.4	23.96	3.76
tricycle	39.84	35.42	25.87	13.87	1.09
awning-tricycle	20.89	19.19	15.48	8.59	1.15
bus	65.19	63.07	58.93	49.91	5.76
pedestrian	59.86	49.15	29.39	8.71	0.29
person	54.85	42.91	23.87	5.86	0.17
bicycle	30.15	23.64	12.0	3.52	0.17
motor	60.69	49.16	29.29	8.71	0.29

TABLE 3. Ablation study on MPFPN. Add: the operation of addition. Concat: the operation of concatenation. SSAM: Supervised spatial attention module.

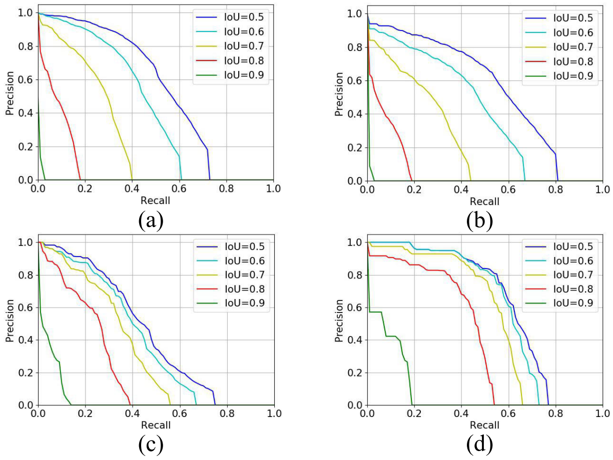
Fusion Strategy	Branch-1	Branch-2	SSAM	AP$_{0.5:0.95}$ [%]	AP$_{0.5}$ [%]	AP$_{0.75}$ [%]
Add	√	√	×	28.05	53.29	25.96
Concat	√	×	×	28.13	53.55	25.88
	×	√	×	27.81	53.19	25.21
	√	√	×	28.32	54.16	25.96
	√	√	√	29.05	54.38	26.99

images are resized to 1440 × 800 pixels, MomentumOptimizer is chosen as the network optimizer, the weight decay is set to 0.0001, and Momentum is 0.9. Meanwhile, we trained a total number of 90k iterations, with the learning rate of 0.001 for the first 70k iterations, 0.0001 for the next 15k iterations, and 0.00001 for the rest of 5k iterations. For data augmentation, randomly flipping image is used in the training stage. In order to match a suitable size of objects in drone-based images, we set the base size of the prior anchor to $\{16, 32, 64, 128, 256\}$ and the anchor ratio is $\{1:2, 1:1, 2:1\}$.

C. COMPARATIVE RESULT IN EACH CATEGORY

We firstly compare all categories to investigate the validity of MPFPN. Cascaded R-CNN with ResNets-101 as the baseline network is structured. Figure 9 gives the visual comparison of MPFPN and FPN, and it can find as a noticeable improvement of our method these objects with a small size. For example, as shown in the red line of Figure 9a, the initial FPN can hardly effectively detect the objects with a tiny area. As a comparison, in Figure 9b, the proposed method can capture and recognize many objects with a few pixels, such as person class depicted in the color green, and the motor class drawn in the color chocolate.

As shown in Table 1, the experiment result suggests that our method reached 29.05% in AP$_{0.5:0.95}$, which is increased by 2.05 points, and the result of each class is better than the initial FPN. It is because the presented MPFPN has more robust feature extraction capability and excellent detection performance. The experiment demonstrates the effectiveness of the method we proposed.



(a) (b) (c) (d)

FIGURE 10. Comparison of the precision-recall curves for different IoU threshold. (a) pedestrian; (b) motor; (c) truck; (d) bus.

As is described in Table 2, the presented method achieves a satisfying performance in most categories when using the lower IoU threshold of 0.5. Meanwhile, we find that the

TABLE 4. Comparison of the state-of-the-art algorithms.

Method	$AP_{0.5:0.95}$ [%]	$AP_{0.5}$ [%]	$AP_{0.75}$ [%]	AR_1 [%]	AR_{10} [%]	AR_{100} [%]	AR_{500} [%]
FPN [27]	25.56	50.21	23.87	0.47	5.07	29.71	39.87
Cascade R-CNN [39]	25.09	49.93	24.02	0.48	4.83	30.38	43.24
DetNet59 [32]	24.28	48.24	23.36	0.45	4.61	29.87	37.28
RefineDet [44]	23.93	46.78	23.08	0.45	4.45	27.14	40.58
RetinaNet [43]	20.84	39.37	19.63	0.41	3.23	23.35	34.27
YOLOv3 [16]	17.15	36.25	14.4	0.64	5.08	21.77	26.04
CornerNet [45]	26.47	52.15	24.78	0.52	5.35	33.38	41.13
ours	29.05	54.38	26.99	0.55	5.81	35.57	45.69

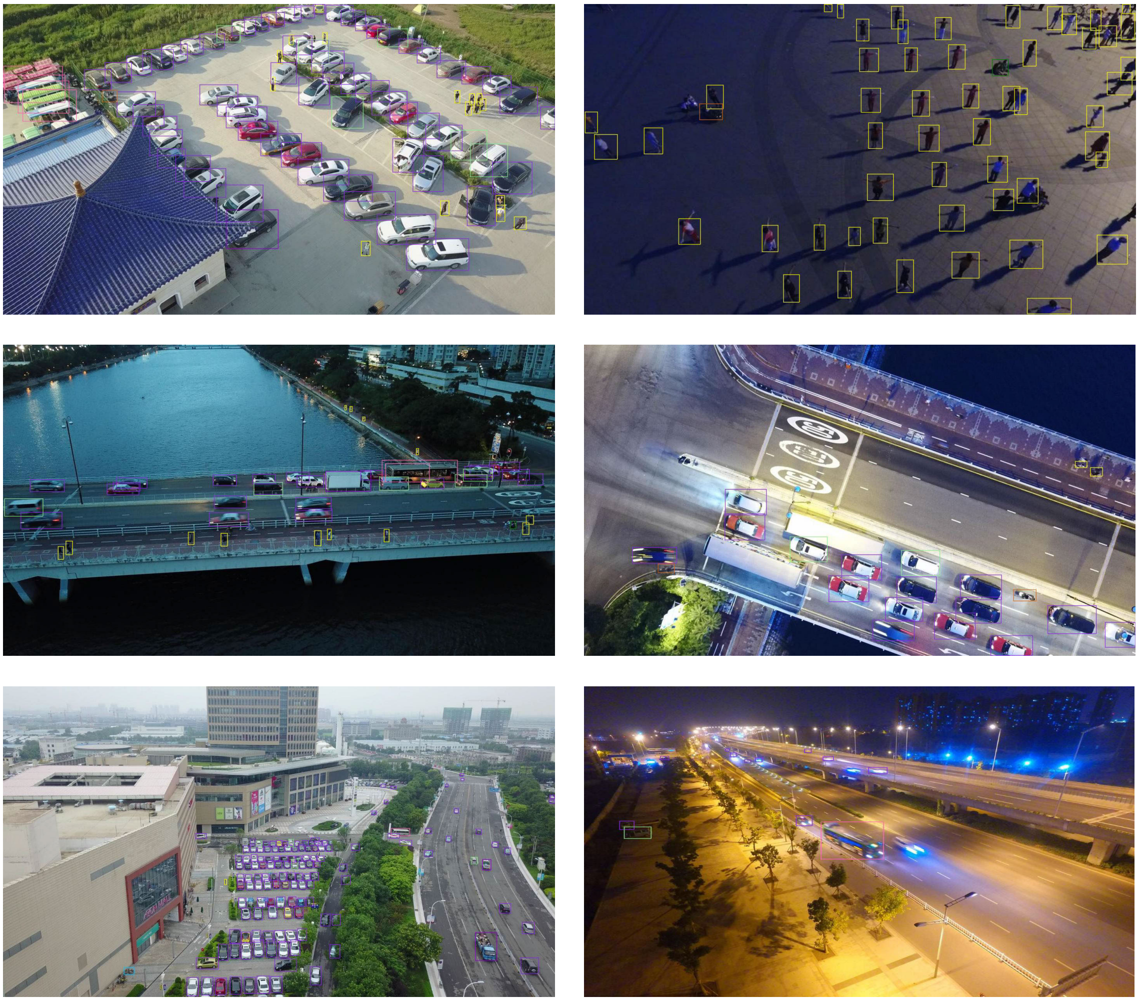


FIGURE 11. Visualization of the proposed framework.

detected results of the above six categories in Table 2 are less affected by the increase of the IoU threshold than the lower four categories. For example, the accuracy in $AP_{0.5}$ and $AP_{0.6}$ of the bus class is declined from 65.19% to 63.07%, while the $AP_{0.5}$ and $AP_{0.6}$ of the person class go sharply decreased from 54.85% to 42.91%, such condition is more obvious in the threshold of 0.7. It suggests that result of some categories is more easily to be affected by the increase of the IoU threshold and thus resulting in a poor performance in the high-IoU threshold detection. Figure 10 plots the precision-recall curves in picked four categories to compare, in Figure 10a and 10b, although the good results are reached in the $AP_{0.5,}$ there is a big shift in the higher threshold for the person class and motor class. One of the possible reasons we guess is that it is quite hard to match the ground-truth of these categories precisely in the process of bounding box regression due to a relatively smaller size to others.

D. ABLATION STUDY

1) EFFECT OF THE PARALLEL BRANCH

In section 3.1, we have analyzed that the parallel branch helps recover feature information of small objects. In Table 3, a comparison is presented to analyze the effect of each parallel branch. As we can see from the table, detection accuracy of $AP_{0.5:0.95}$ is increased by 1.13 points when only applying parallel branch-1 and 0.81 points when only using parallel branch-2. Results indicate that these two parallel branches are both effective in providing more sufficient and useful feature representations.

2) FUSION STRATEGY

Comparative results of two different fusion methods of feature maps are also presented in Table 3. We can see that the addition operation reaches 28.05% in $AP_{0.5:0.95}$, the Concatenation operation is increased to 28.32%. It suggests that the fusion method of concatenation can achieve a higher detection accuracy, but more computing resources are also occupied when applying this method in networks.

3) EFFECT OF SSAM

Based on the operation of using parallel branch and concatenation, we further evaluate the effect of attention module, as shown in Table 3, $AP_{0.5:0.95}$ is increased from 28.32% to 29.05%, which is improved by 0.73 points. This shows that using SSAM can weaken the impact of background noise and enhance the representation of objects.

E. COMPARING WITH SOME STATE-OF-THE-ART ALGORITHMS

We make a comparison with the existing popular object detection method on the validation set of VisDrone. As shown in Table 4, the experiment result of our method suggests a state-of-the-art performance on both AP and AR compared with others. Specifically, we reach 29.05% in the $AP_{0.5:0.95}$ and 45.69% in the AR_{500}. The visualization results of the proposed framework suggest that our method has a competitive performance, as show in Figure 11. Besides, in the competition of the VisDrone-2020 challenge, our method achieves 22.85% and ranks 38th in all 85 participating teams, the leaderboard of the VisDrone-2020 challenge has been released in http://aiskyeye.com/leaderboard/. Note that we don't use any image augmentation tricks or model fusion methods due to the limitation of GPU memory or other hardware.

V. DISCUSSIONS

Many groups of experiments have been presented to verify the validity of the proposed method. The advantage of our framework can be summarized as follow: (1) MPFPN has a better capability to extract sufficient and contextual semantic information for the discriminative representation of the objects with a small size. (2) Cascade architecture implements a high-quality bounding box regression, which provides a precise localization of objects in UAV-captured images. However, there are still some deficiency existing in the proposed networks, which can be attributed to two aspects: unbalanced training samples and false alarm.

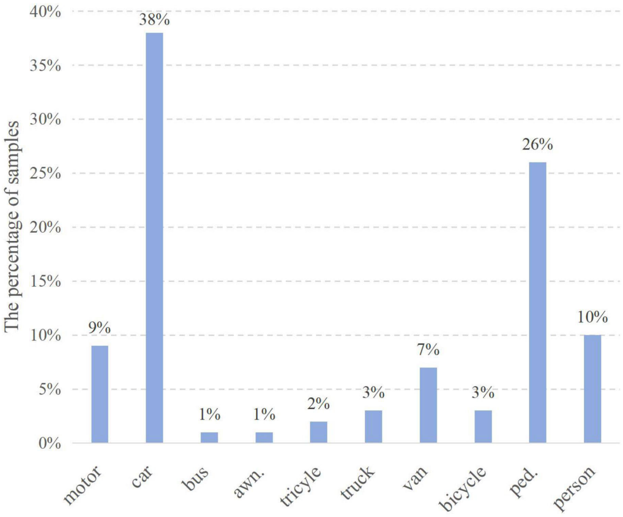


FIGURE 12. The percentage of the training samples. Categories awn. refers to awning-tricycle, ped. refers to pedestrian.

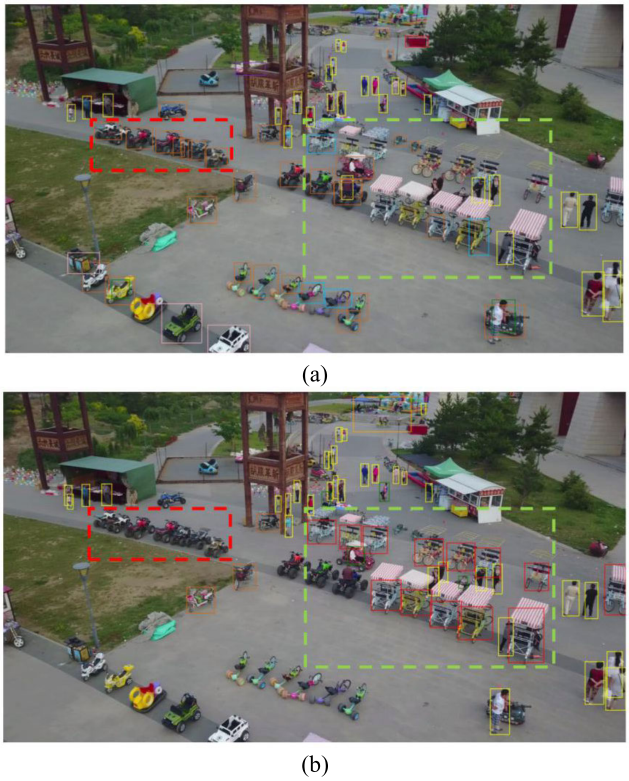


FIGURE 13. Visual comparison. (a) detected result in the proposed method; (b) Ground-truth. The Red dashed line refers to the confused objects. The Green dashed line refers to the categories with a lack of training sample.

A. UNBALANCED TRAINING SAMPLES

Although the proposed structure achieves a satisfying performance in most category, we can see from above Table 1 that the detection accuracy in some classes is extremely lower than others. For example, the detected result of the car class produces 57.85% in $AP_{0.5:0.95}$, while only reaches 12.01% and 12.41% on awning-tricycle and bicycle. One of the possible reasons is that there exists an unbalanced number of trained samples. As shown in Figure 12, awning-tricycle and bicycle occupy 1% and 3% respectively in all samples, in Figure 13a, it can be seen an undesirable visual result to class awning-tricycle, which is drawn by the green dashed line. To address this issue, A data augmentation strategy for small data will be considered to promote detection performance in future work.

B. FALSE ALARM

Another issue that weakens the proposed detection networks is the incorrect recognition when detecting the objects with a similar appearance to the ground-truth. In Figure 13a, the objects inside the red dashed line are detected to the class motor, but it does not belong to the training samples (see figure 13b). One possible reason is that such objects or scenario never appear in the training images; there exists a big gap in the testing set and training set. To deal with this issue, we consider adopting the training strategy to learn the specific characteristic of some interferential objects on the testing images or using the trained model on the other datasets for removing it in the post-processing stage.

VI. CONCLUSION

In this article, considering the difficulty of detecting small objects, we have proposed an object detection framework for UAV-captured images, many novel methods were presented in this model. For example, the proposed MPFPN utilizes two additional parallel branches to obtain sufficient feature representation of small objects, based on that, a supervised spatial attention module named SSAM is adapted to restrain the noise interference and highlight the object information effectively. Meanwhile, to enhance the localization capability of objects in UAV-captured images, we took cascade architecture in the Fast R-CNN stage. Extensive experiments were performed on the drone-based datasets named VisDrone-DET, and the results demonstrate that the presented framework has a competitive effect and reached a state-of-the-art performance in object detection in UAV aerial images. Despite reaching a desirable performance in most classes, there are still some issues existing in the framework, such as unbalanced training samples and incorrect recognition of the objects that never showed up in the training images. In the future, we need to find out the solution to solve the existed problems in our framework.

REFERENCES

[1] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote Sens.*, vol. 4, no. 6, pp. 1671–1692, Jun. 2012, doi: 10.3390/rs4x061671.

[2] L. Wang, Y. Lan, Y. Zhang, H. Zhang, M. N. Tahir, S. Ou, X. Liu, and P. Chen, "Applications and prospects of agricultural unmanned aerial vehicle obstacle avoidance technology in China," *Sensors*, vol. 19, no. 3, p. 642, Feb. 2019, doi: 10.3390/s19030642.

[3] Y. Yang, Z. Lin, and F. Liu, "Stable imaging and accuracy issues of low-altitude unmanned aerial vehicle photogrammetry systems," *Remote Sens.*, vol. 8, no. 4, p. 316, Apr. 2016, doi: 10.3390/rs8040316.

[4] J. Eun, B. D. Song, S. Lee, and D.-E. Lim, "Mathematical investigation on the sustainability of UAV logistics," *Sustainability*, vol. 11, no. 21, p. 5932, Oct. 2019, doi: 10.3390/su11215932.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[8] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5353–5360, doi: 10.1109/CVPR.2015.7299173.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Sep. 2010.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[12] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[19] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06559*. [Online]. Available: https://arxiv.org/abs/1701.06559

[20] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, and H. Cheng, "VisDrone-DET2018: The vision meets drone object detection in image challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. (ECCV) Workshops*, Munich, Germany, Sep. 2018, pp. 437–468.

[21] S. Zhuang, P. Wang, B. Jiang, G. Wang, and C. Wang, "A single shot framework with multi-scale feature fusion for geospatial object detection," *Remote Sens.*, vol. 11, no. 5, p. 594, Mar. 2019, doi: 10.3390/rs11050594.

[22] W. Ma, Q. Guo, Y. Wu, W. Zhao, X. Zhang, and L. Jiao, "A novel multi-model decision fusion network for object detection in remote sensing images," *Remote Sens.*, vol. 11, no. 7, p. 737, Mar. 2019, doi: 10.3390/rs11070737.

[23] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network," *Remote Sens.*, vol. 12, no. 9, p. 1432, May 2020, doi: 10.3390/rs12091432.

[24] C. Chen, J. Zhong, and Y. Tan, "Multiple-oriented and small object detection with convolutional neural networks for aerial image," *Remote Sens.*, vol. 11, no. 18, p. 2176, Sep. 2019, doi: 10.3390/rs11182176.

[25] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware ReAssembly of FEatures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 3007–3016, doi: 10.1109/ICCV.2019.00310.

[26] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," 2017, *arXiv:1703.06211*. [Online]. Available: http://arxiv.org/abs/1703.06211

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016, *arXiv:1612.03144*. [Online]. Available: http://arxiv.org/abs/1612.03144

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[31] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," 2016, *arXiv:1605.06409*. [Online]. Available: http://arxiv.org/abs/1605.06409

[32] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," 2018, *arXiv:1804.06215*. [Online]. Available: http://arxiv.org/abs/1804.06215

[33] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2017, *arXiv:1709.01507*. [Online]. Available: http://arxiv.org/abs/1709.01507

[34] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 8231–8240, doi: 10.1109/ICCV.2019.00832.

[35] S. Woo, J. Park, J. Lee, and I. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[36] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," 2019, *arXiv:1901.03278*. [Online]. Available: http://arxiv.org/abs/1901.03278

[37] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*. [Online]. Available: http://arxiv.org/abs/1706.09579

[38] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," 2017, *arXiv:1703.01086*. [Online]. Available: http://arxiv.org/abs/1703.01086

[39] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 28, 2019, doi: 10.1109/TPAMI.2019.2956516.

[40] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6053–6062, doi: 10.1109/ICCV.2019.00615.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[42] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," 2018, *arXiv:1804.07437*. [Online]. Available: http://arxiv.org/abs/1804.07437

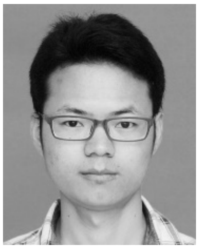
[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[44] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4203–4212, doi: 10.1109/CVPR.2018.00442.

[45] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

YINGJIE LIU received the B.S. degree in electronic information engineering from Tianjin Polytechnic University, in 2015. He is currently pursuing the M.E. degree with the North University of China, Taiyuan. His current research interests include machine learning and computer vision.

FENGBAO YANG received the Ph.D. degree in measurement technology and instrument from the North University of China, Taiyuan, China, in 2003. From 2004 to 2007, he was a Postdoctoral Research Fellow with the Beijing Institute of Technology. He is currently a Full Professor with the North University of China. His current research interests include information fusion, performance detection and evaluation of complex systems, and information fusion theory of uncertainty. He presided more than two items of the National Natural Science Foundation.



PENG HU received the B.S. degree in physics from the North University of China, Taiyuan, China, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include image processing and artificial intelligence.