A Novel Object Detection Method in City Aerial Image Based on Deformable Convolutional Networks

Li-Qun Zhou, Peng Sun, De Li and Jin-Chun Piao

Department of Computer Science and Technology, Yanbian University, Yanji, China
Corresponding author: Jin-Chun Piao (jcpiao@ybu.edu.cn).

This research was supported by Education Department of Jilin Province of China (grant number JJKH20210570KJ) and the National Natural Science Foundation of China (grant number 62062064).

ABSTRACT Unmanned Aerial Vehicle (UAV) has been widely applied for aerial object detection. Recently, this issue has become a research hot spot in the field of computer vision. Currently, the performance of conventional methods to detect small objects has reached a breakthrough point. Moreover, the dense distribution and large-scale variation of objects in aerial images significantly affect the detection accuracy. In order to resolve this problem, a novel structure based on the YOLOv3 is proposed in the present study. To this end, the backbone network is replaced by ResNet_vd50 to prevent information loss in the downsampling process. Then the backbone network is modified by deformable convolution to improve the detection ability of deformed objects. In this regard, SE attention is embedded in the ResNet_vd block to improve the expression ability of features. Furthermore, the Soft-NMS algorithm is introduced for bounding box fusion to resolve the occlusion problem. Finally, the MixUp method is used in the data augmentation stage to enrich the background information by fusing different images. Based on the obtained results, it is concluded that the proposed method has higher accuracy in aerial images than state of art object detection methods.

INDEX TERMS Aerial images, Object detection, Deformable convolution, Attention mechanism, Fusion strategy

I. INTRODUCTION

OBJECT detection on UAV perspective has become a research hot spot in the computer vision field in the past few years. In this regard, numerous technologies have been widely applied in the construction of smart cities. For example, under pandemic conditions, mass gatherings may increase the spread of the epidemic. In order to resolve this problem, pedestrian detection through UAV can estimate the crowd congestion, thereby assisting epidemic prevention [1]. As another example, based on new traffic rules, cyclists must wear a helmet when riding an e-bike. In this regard, e-bike detection techniques can be combined with helmet detection technologies to realize automatic detection through UAVs [2]. Moreover, in case of violation, UAVs can record short videos of the incident and send them to a nearby base station to provide data for authorities to take legal actions [3].

Meanwhile, object detection has attracted many scholars for a long time. In the field of computer vision, conventional object detection algorithms are generally based on statistical machine learning [4]. In these algorithms, it is necessary to extract features from the region proposal in a sliding way and then classify them. The main drawback of this method is the high complexity of the algorithm, originating from a large number of sliding windows for traversal, and challenges to manually design a robust feature extractor. Accordingly, conventional object detection algorithms can’t meet the requirements in practical applications.

With the rapid progress of deep learning technology in the past few years, object detection algorithms represented by a CNN (Convolutional Neural Network) have achieved satisfactory results from different aspects [5]. The main characteristic of CNN is to use a large number of samples to learn image features. Accordingly, its features have promising generalization ability to perform detection tasks. Therefore, numerous CNN schemes including SSD series [6], [7], YOLO series [8]–[12], and CenterNet series [13], [14], have been proposed. However, as shown in Figure 1, objects from
aerial image are different from those from common horizontal photographic image. A large number of small objects exist in aerial image.

Motivated by this observation, different methods [15]–[19] have been proposed to detect small objects. However, detecting objects in aerial images using a deep learning algorithm has other challenges. For example, many objects may exist in aerial images, resulting in occlusion problems between objects. Meanwhile, the complex and diverse background of aerial images leads to the difficulty of feature extraction. Different scales originating from different shooting angles and distances of the same object are also a challenge. UAVs usually have high speeds. In some cases, imaging at high moving speeds may result in blur images and geometric deformation, which makes it difficult to detect objects. In the face of these problems, network designs aiming at only small objects will not be able to obtain full object information on the aerial image. Accordingly, it is necessary to improve the feature extraction network of the detector, especially the way of convolution, thereby improving the ability to detect objects in aerial images.

In response to the above ideas, it is intended to propose a novel detection algorithm for aerial images based on the YOLOv3 algorithm [10], and provide methodological guidance for other object detection algorithms to improve the detection ability of small objects in aerial images. To this end, ResNet_vd50 is initially applied to replace Darknet53, solve the information loss and improve the detection speed. Second, in stage 3 of the ResNet_vd50, a $3 \times 3$ convolution is replaced with a deformable convolution to improve the learning ability of deformed objects. Moreover, in the backbone network, the residual block embedded with the attention mechanism is used to further improve the feature extraction ability. Finally, the Soft-NMS (Soft Non Maximum Suppression) algorithm is applied to resolve the occlusion problem in aerial images. The main contributions of the present study can be summarized as follow:

1. Based on YOLOv3, a novel deep learning-based algorithm is proposed for object detection in aerial images.
2. The ResNet_vd50 is redesigned by adding deformable convolution and attention mechanism as the backbone network to fully extract the feature information of deformed objects in aerial images.
3. NMS is replaced with Soft-NMS to resolve the occlusion problem in aerial images.
4. The MixUp method is used to enrich the background information and enhance the detector generalization ability.

This article is organized as follows: Related works on general and aerial object detection methods are reviewed in Section II. Then the proposed method of aerial object detection is presented in Section III. Moreover, experiments and the results are discussed in Section IV. Finally, the main achievements and conclusions are summarized in Section V.

II. RELATED WORK

Object detection is an important research direction in the field of deep learning [20]. Before the promotion of deep learning, conventional object detection methods were used in this regard. These methods consist of 3 steps, including object location, feature extraction, and object classification. Object location is realized through the sliding window with a fixed size and shape. It slides over the image in a specified step to obtain a fixed-size area. However, it is a very time-consuming process to find a perfect window for each object so that providing many redundant windows is unavoidable, which reduces object positioning speed. It should be indicated that feature extraction is the most important task in object detection. After obtaining the object region through the sliding window, it is necessary to analyze the semantic and visual representation of each object in the image. This can be realized through different feature descriptors such as SIFT [21], HOG [22], and LBP [23]. Then the object region is classified by the feature descriptor and the bounding box of the object can be drawn. Conventional classifiers in this regard are AdaBoost [24], SVM [25] and Random Forest [26]. However, since the conventional methods can’t effectively detect large-scale data, a bottleneck appeared in 2010. At this time, CNN has made a breakthrough in the field of object detection. This scheme can be divided into one-stage and two-stage detectors.

One-stage detectors directly predict the category and location of the object without generating the region of the proposal. Accordingly, it has a high speed but low accuracy. YOLO [8] introduced the regression method into the object detection framework. In this regard, the image was divided into a $7 \times 7$ grid, and features were extracted through CNN. Then the confidence and location of objects were directly regressed in each grid. However, the anchor box is not used in this method so that it has a relatively low location accuracy. In order to resolve this problem, Wei Liu et al. proposed SSD to set the anchor box on the feature maps with different scales, and then directly made the anchor classify and regress [6]. To eliminates the main obstacle impeding one-stage detector from achieving state-of-the-art accuracy, K.He et al.propose Focsl loss function [27]. To improve the detection speed, YOLOv2 [9] was proposed. In this scheme, the connecting layer was removed completely and the anchor box was used to limit the box prediction. Studies show that the detection speed and accuracy of YOLOv2 on the VOC 2007 are about 64 FPS and 76.8%, respectively. Subsequently, YOLOv3 [10] was proposed and greatly improved the detection speed and accuracy by introducing residual structure and feature pyra-
mid structures. Tian et al. proposed FCOS [28] to directly predict the possibility of object existence and the bounding box coordinates without anchor reference. Alexey. B et al. proposed YOLOv4, where the performance of the detector was improved by modifying the backbone network and using data augmentation methods [11]. However, this method was not customized to detect aerial images. In order to resolve these shortcomings, the main objective of the present study is to improve the object detection performance in aerial images by developing a general YOLOv3 algorithm. This idea is expected to provide an effective way to develop higher-order algorithms, including YOLOv4, YOLOv5, and improve the ability to detect small objects.

The two-stage detectors generate the region of the proposal at first and then refine them in the next stage. Consequently, the detected speed of this method is generally lower than that of one-stage detectors. In this regard, Ross et al. proposed the first CNN-based object detection framework called R-CNN [29].

Studies demonstrate that CNN features have greater discrimination between objects and backgrounds compared to hand-craft features [30]. Kaiming He proposed SPPNet (Spatial Pyramid Pooling Network) [31]. In this feature, features should be extracted only once for the whole image, and then the region proposal can be mapped to the conv5 layer. Based on SPPNet, excellent two-stage detectors such as Fast R-CNN [32] and Faster R-CNN [33] have been established to improve the ROI (Region of Interest) pooling layer and RPN (Region Proposal Network), respectively. Further investigations in this regard show that these schemes can greatly improve the detector accuracy. Based on the Faster R-CNN, Joseph et al. used an energy-based classification head and unknown-aware RPN to identify unknown objects. Furthermore, contrastive learning was performed in the feature space to learn discriminative clusters and add new classes in a continual [34].

Different from general object detection, aerial object detection faces additional challenges. First, there are many small objects in an aerial image. Second, since the shooting angle may change greatly, the object shape also changes. Finally, there exist object occlusion problems in aerial images. In order to resolve these challenges, many investigations have been conducted so far. In this regard, Zhang et al. used a scale-adaptive network to improve the detection performance of small objects. They also improved the detection speed on mobile platforms [35]. Moreover, Yang et al. proposed a network that uses customized anchors to deal with the scale changes originating from the viewpoint [36]. Wang et al. proposed a cluster network to generate regions of dense objects. To this end, a scale network was applied to adjust the shape of the resulting region [37]. Chen et al. studied the learning methods of low-scoring regions from the detector and scoring regions and improved low-level features, thereby improving the detection accuracy of small objects [38]. Pang et al. proposed an aerial image detection method based on the convolutional neural network [39]. In this method, object detection was combined with semantic segmentation to improve the detection performance.

Although promising results have been achieved through these methods, the problem of poor object detection accuracy in aerial images caused by object deformation has not yet been considered. To resolve this problem in the present study, it is intended to analyze the reasons for insufficient detection accuracy. In this regard, an aerial object detection method is proposed based on YOLOv3 and existing object detection technology.

### III. PROPOSED METHOD

Figure 2 shows the architecture of the proposed model. When the image is processed, it is fed to ResNet_vd50, which is easier to optimize. In each residual block, an SE (Squeeze-and-Excitation) attention is embedded to improve the feature representation ability of objects in aerial images. Then the feature maps generated by C3, C4, and C5 are used to form the FPN (Feature Pyramid Network). With the deepening of the number of network layers, each pixel in the feature...
layer has a larger receptive field and is responsible to detect large objects. Therefore, a deformable convolution is applied in the shallow C3 layer to enhance the feature extraction of a deformed small object in aerial images. Finally, multi-scale object detection is performed through three detection heads, and the bounding boxes are fused by the Soft-NMS algorithm before the preparation of the detection results. Applying the proposed network structure design results in independent function in each part, thereby facilitating subsequent improvements.

**A. BACKBONE NETWORK CONSTRUCTION**

![Diagram](image)

**FIGURE 3.** Comparison of ResNet(l) and ResNet_vd(r)

1) **ResNet_vd**

Generally, the deeper the network, the stronger the ability to extract features, the larger the receptive field of view, and the more advanced the semantic information. However, the gradient disappears as the network deepens. Consequently, the neural network can’t be simply stacked. The ResNet is an effective way to solve the problems of disappearing gradient by identity mapping, making it possible to train hundreds of layers of the network.

According to the number of layers, ResNet can be divided into five categories where ResNet50 with a 50-layer structure is the most common scheme. This mainly originates from better feature extraction ability, fewer parameters, and more flexibility of ResNet50 compared with Darknet53. Accordingly, the ResNet50 is used in the present study as the backbone network. It is worth noting that the idea of ResNet_vd [40] is also used in this article. Figure 3 indicates that in the non-identity branch, the down-sampling is adjusted from the 1×1 convolution from the beginning to the 3×3 convolution in the middle, thereby preventing the loss of information. In the identity branch, the average pooling is added to complete the down-sampling operation. Finally, the feature maps of two branches are added by pixel.

2) **Deformable Convolution**

There are many small objects in the aerial images, which have deformation and blur problems. Some deformations will pass through the marked area and cause information loss, resulting in deviations in the regression of the bounding box. In the backbone network, the feature map of C3 is responsible for detecting small objects. In this regard, the conventional convolution is modified in this layer using deformable convolution [41].

Different from the fixed geometry of conventional convolution, deformable convolution adds an offset variable to the position of each sampling point in the convolution kernel. By learning these variables in an unsupervised manner, the convolution kernel can randomly perform sampling near the current position instead of regular grid points. Each point in the convolution kernel can be scaled and changed to find the most suitable receptive field so that the network can focus on the object deformation to extract more object features, thereby improving the detection performance.

![Figure 4](image)

**FIGURE 4.** (a) conventional convolution; (b)-(d): deformable convolutions.

Figure 4 shows the conventional convolution and deformable convolutions, where (a) is the conventional 3×3 convolution kernel, and the region of orange sampling dots is the receptive field of conventional convolution, and it can be seen that it is a 3×3 rectangle. (b)-(d) are three cases of deformable convolution, blue dots are obtained by adding learnable offsets (blue arrow) on the basis of conventional sampling points, therefore, convolution kernels can adapt to objects of different shapes and scales. The deformable convolution can be mathematically expressed in the form below:

\[
y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{1}
\]

\[
x(p) = \sum_{q} G(q, p) \cdot x(q) \tag{2}
\]

For the value of each position \(p_0\) on the output feature map \(y\), the conventional convolution is a regular sampling in the box, while the deformable convolution adds the offset \(\Delta p_n\) to the original sampling position, so the new sampling position is \(p_n + \Delta p_n\). Usually, the offset is a float that should be solved by bilinear interpolation. Where \(p\) represents \(p_0 + p_n + \Delta p_n\) for Equation 2, \(q\) represents all integer spatial locations in the feature map \(x\), \(G(q, p)\) is a two-dimensional bilinear interpolation kernel function, which can be decomposed into two one-dimensional kernel functions.

Figure 5 shows a deformable convolution used in stage 3 of the backbone network. The input and the output feature maps have the same size, and the convolution kernel size is 3×3. In order to learn the offset, another 3×3 convolutional
layer with 2N channels is defined, representing the offsets of N points along x and y directions. During the training process, the output feature map and the offset are learned simultaneously.

3) ResNet_vd with SE
The complex background information of aerial images can improve the generalization ability of the model. On the other hand, it will also interfere with the detection ability of the model. To further improve the detection ability of the proposed model, SE attention mechanism [42] is added to obtain the dependency correlation between features and improve the expressive ability of features.

\[
\begin{align*}
    z_c &= F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j), \quad z \in R^C \\
    s &= F_{ex}(z_c, W) = \text{Sigmoid}(W_2 \text{ReLU}(W_1 z_c))
\end{align*}
\]

After obtaining the global feature \( z_c \), the "excitation" operation is performed according to Equation 4. First, the C-dimension vectors \( z_c \) are fully connected to get \((C/r)\)-dimension vectors. Then ReLu activation function is used followed by a full connection, and the \((C/r)\)-dimension vector is converted into the C-dimension vector. Finally, the Sigmoid activation function is used to fit the value in the range \((0,1)\) and the final output of the block is obtained in the form below:

\[
\tilde{x}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c
\]

Equation 5 indicates that the final output \( \tilde{x}_c \) can be obtained by rescaling \( u_c \) with the activation parameters \( s_c \).

B. BOUNDING BOX DECODER

Figure 7 shows the flowchart of the ResNet_vd block with an SE attention mechanism on the non-identity branch. Since the convolution operation is performed in a local space, it is an enormous challenge to obtain enough data to extract the correlation between the channels. Therefore, the "squeeze" operation is applied to encode the spatial features on the channel as a global feature \( z_c \). This feature, which has a global receptive field, can be mathematically expressed as follows:

\[
z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j), \quad z \in R^C
\]

where \( u_c \) denotes the c-th channel of the output feature map, and \( H, W, C \) are the height and width of the output feature map and the number of channels, respectively. Moreover, \( R^C \) represents the channel characteristics.

Since the output of the model are parameters that adjust the position of the anchor box, decoding is required to get the location of the bounding box. As shown in Figure 7, \( t_x, t_y, t_w, t_h \) are coordinates of the bounding box. Moreover, \( b_x \) and \( b_y \) denote the center of the bounding box, \( b_w \) and \( b_h \) are its length and width, respectively. If the grid cell is offset from the top left corner of the image by \( (C_x, C_y) \) and the anchor box has width and height of \( p_w, p_h \), then the decoding equation can be expressed in the form below:

\[
\begin{align*}
    b_x &= \text{Sigmoid}(t_x) + C_x \\
    b_y &= \text{Sigmoid}(t_y) + C_y \\
    b_w &= p_w \times e^{t_w} \\
    b_h &= p_h \times e^{t_h}
\end{align*}
\]

C. BOUNDING BOX FUSION

Each ground truth has many corresponding bounding boxes, but only one bounding box can be used. Accordingly, it is necessary to apply the bounding box fusion. For example,
Figure 8 illustrates samples of fusion bounding boxes for some objects. Unlike the traditional NMS algorithm that forces the scores of adjacent bounding boxes to zero, the Soft-NMS [43] is applied in the present study to fuse the bounding box and resolve the occlusion problem.

\[
s_i = \begin{cases} 
    s_i, & \text{IoU}(b_1, b) < N \\
    s_i(1 - \text{IoU}(b_1, b)), & \text{IoU}(b_1, b) \geq N 
\end{cases} \quad (10)
\]

Equation 10 indicates that Soft-NMS would decay the scores of detections above a threshold \( N \) as a linear function. Consequently, detection boxes far from \( b \) would not be affected and those near \( b \) would be assigned a greater penalty.

**Input:** \( B = \{b_1, \ldots, b_N\}, S = \{s_1, \ldots, s_N\}, N_t \)
- \( B \) is the list of initial detection boxes
- \( S \) contains corresponding detection scores
- \( N_t \) is the NMS threshold

**Algorithm of Soft-NMS**

1: \( D \leftarrow \{\} \)
2: while \( B \neq \text{empty} \) do
3:     \( m \leftarrow \text{argmax}_i \sum_s S_i \)
4:     \( M \leftarrow b_m \)
5:     \( D \leftarrow D \cup M; B \leftarrow B - M \)
6:     for \( b_i \in B \) do
7:         \( s_i \leftarrow s_i \cdot \text{iou}(M, b_i) \)
8:     end for
9:     if \( s_i \leq N_t \) then
10:         \( B \leftarrow B - b_i; S \leftarrow S - s_i \)
11: end if
12: end while
13: return \( D \), \( S \)

Figure 9 shows the process of Soft-NMS, indicating that the process begins with a list of detection boxes \( B \) with scores \( S \). After selecting the bounding box with the maximum score \( M \), it is subtracted from the set \( B \) and appends it to the set of final detections \( D \). It also removes any box with \( s_i \) less than the threshold \( N_t \). This process is repeated for all remaining boxes.

Figure 10 shows the effect of Soft-NMS on the obtained results. It is observed that the masked object does not remove, but reduces the score to allow it to participate in the next fusion, thereby reducing the impact of the occlusion problem on the detection result.

**D. LOSS FUNCTION**

It should be indicated that the total loss originates from bounding box coordinate predictions, confidence predictions, and object classification. This can be mathematically expressed in the form below:

\[
\text{loss} = l_{\text{box}} + l_{\text{obj}} + l_{\text{cls}} \quad (11)
\]

Each term in Equation (11) is discussed below.

\[
l_{\text{box}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{\text{obj}} (2 - w_i \times h_i) \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right] \quad (12)
\]

Equation 12 indicates that the bounding box loss consists of the center point loss and the loss caused by predicting width and height. Moreover, \( x_i \) and \( y_i \) are the coordinates of the center point of the \( i \)-th ground truth relative to the grid cell, and \( \hat{x}_i \) and \( \hat{y}_i \) are the corresponding predicted values; When the \( j \)-th grid cell contains an object, \( I_{i,j}^{\text{obj}} = 1 \), otherwise \( I_{i,j}^{\text{obj}} = 0 \); \( w_i \) and \( h_i \) denote the true scaling of width and height; \( \hat{w}_i \) and \( \hat{h}_i \) represent the corresponding predicted values; \( S \) is the number of grid cells, and \( B \) is the number of anchor boxes. The term \((2 - w_i \times h_i)\) is used to increase the proportion of small targets [6].

The confidence loss can be expressed as follows:

\[
l_{\text{obj}} = \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{\text{noobj}} (c_i - \hat{c}_i)^2 + \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{\text{obj}} (c_i - \hat{c}_i)^2 \quad (13)
\]

When the \((i, j)\) grid cell does not contain an object, \( I_{i,j}^{\text{noobj}} = 1 \), \( I_{i,j}^{\text{obj}} = 0 \), otherwise \( I_{i,j}^{\text{noobj}} = 0, I_{i,j}^{\text{obj}} = 1 \); Meanwhile, \( c_i \) is the confidence of the \( i \)-th ground truth (0 or 1) and \( \hat{c}_i \) is the confidence of the bounding box.

\[
l_{\text{cls}} = \sum_{i=0}^{S^2} I_{i,j}^{\text{obj}} \sum_{c \in \text{classes}} \left[ p_i(c) \log(\hat{p}_i(c)) + (1 - p_i(c)) \log(1 - \hat{p}_i(c)) \right] \quad (14)
\]

Equation 14 indicates that cross-entropy is used to calculate the classification loss. Only when \( I_{i,j}^{\text{obj}} = 1 \), the bounding box generated by the \( j \)-anchor box of the \( i \)-th grid cell will calculate the classification loss. \( \hat{p}_i \) is the presence...
probability of the object in the training sample, and $p_i$ is the corresponding prediction probability of the model.

IV. EXPERIMENTS AND THE RESULTS

A. DATASET

To evaluate the performance of the proposed method, it is applied to the VisDroneDet2018 dataset [44], which contains 10,209 images captured by drones. Among these images, 6,471 images are considered for training, 548 images for validation, and 3,190 images for testing. During the evaluation, 10 common objects, including pedestrians, people, bicycles, cars, vans, and buses, are involved. Figure 10 shows that the VisDroneDet2018 dataset includes many small objects, and there are problems such as occlusion and blur. Based on the performed information analysis about objects, some labels are recognized. Accordingly, people and pedestrians are set as people, tricycles and awning tricycles are set as a tricycle, and cars and vans are set as car. The test set includes the test-dev and test-challenge. Since the test-challenge is only available to contestants, the test-dev is used. It should be indicated that the distribution of these images is similar to the training set, which can effectively reflect the training effect of the model.

B. DATA AUGMENTATION

To improve the generalization ability of the model, the MixUp method [45] is applied in the present study to enrich the background information. This method fusions two images with a certain ratio, thereby increasing the number of samples and effectively improving the prediction ability of the model. Meanwhile, the MixUp method can reduce the fluctuations during training. Figure 12 shows an improved image of the street and basketball court using the MixUp method with different fusion ratios. It is observed that the best object and background information on the two images can be achieved when the fusion ratio is set to 0.5.

C. IMPLEMENTATION DETAILS

All of the experimental results are performed on the VisDroneDet2018 dataset. To this end, Paddle 1.8.4 is used as the deep learning framework, and ImageNet is used as the pre-training model to initialize the network. Moreover, the size of input images is randomly selected from 320 × 320 to 608 × 608 pixels to increase the network robustness. The Gradient Descent with Momentum is selected as the optimizer, and the regularization factor and the Momentum are set to 0.0005 and 0.9, respectively. Meanwhile, the original data are clustered to obtain the best anchor. It is worth noting that the model is trained on a single GPU (Tesla V100 32GB) with LR warm-up technology to gradually increase to 0.01, and the batch size is set to 32.

D. ABLATION STUDY

In the present study, a series of ablation experiments are carried out on the VisDroneDet2018 dataset. The obtained results in this regard are presented in Table 1, where the first row shows the baseline performance representing the original YOLOv3. Then a variety of technologies are combined based on the baseline to verify the effects of different combinations on the model. Each combination is named appropriately. For example, YOLOv3_Rvd means to replace the backbone of YOLOv3 with Resnet50_vd, YOLOv3_Rvd_MU means to use the MixUp method based on YOLOv3_Rvd. Based on the performed optimization, it is found that the mAP of the proposed method is 4.8% higher than that of the baseline, while the operating speed is 4 FPS faster.

1) YOLOv3_Rvd

In this module, the backbone network is changed from Darknet53 to Restnet50_vd. Since the best anchor box is used and resnet50_Rvd is more concise than Darknet53, it is 11 higher than the baseline FPS and 1.2% higher than mAP. Moreover, AP@.75 and AP@L are improved by 1.7% and 3.1%, respectively. Although changing the backbone network reduces the AP@S up to 0.5%, these losses will be optimized back later.
2) YOLOv3_Rvd_MU

Based on the previous stage, the MixUp method is used in this module to fuse different images. When there are few samples, this method increases the number of samples to prevent overfitting. Moreover, when there are many samples, this method enriches the background information to improve the generalization ability. Therefore, compared with the YOLOv3_Rvd, mAP increases by 1.3%. Furthermore, AP@.75 and AP@L increase by 2.5% and 2.8%, respectively. However, the speed decreased by 2 FPS, which may be attributed to the data processing before training.

3) YOLOv3_Rvd_MU_SE

In this module, the SE attention mechanism is added to the YOLOv3_Rvd_MU to improve the feature expression ability. Then the improved feature is transmitted to the C3 layer through the FPN to improve small object detection ability. Accordingly, mAP and AR@L increased by 3.4% and 10%, respectively. However, applying this module increases the computational complexity, thereby reducing the speed by 2 FPS.

4) YOLOv3_DCN

To demonstrate that the deformable convolution can improve the model performance, only the deformable convolution is replaced in the C3 layer of the backbone network of YOLOv3. In this case, mAP and AR@L increased by 3.4% and 10%, respectively, indicating that applying the deformable convolution can effectively reduce the missed detection rate of the model for deformed objects in aerial images. However, the computational complexity of deformable convolution is higher than that of conventional convolution, the detection speed is reduced by 3 FPS.

5) YOLOv3_MU

To evaluate the effectiveness of the data enhancement method, the MixUp method is used on the baseline. The obtained experimental results show that mAP is increased by 1.8%, while only reducing 2 FPS, further revealing that the MixUp method can be used with almost no additional computational overhead.

6) YOLOv3_Rvd_SE

In this technical combination, the role of SE in improving the model is analyzed. To this end, SE is embedded in the model, while ResNet50_vd is used as the backbone. The obtained results show that compared with YOLOv3_Rvd, mAP is increased by 1.1%, indicating that the introduction of the backbone with SE can further extract features and enhance the detection performance of the model.

7) YOLOv3_Rvd_MU_SE_DCN

In this module, all previously mentioned techniques are used. Considering large viewing angles and long-distance of UAV perspective, small objects in aerial images are usually deformed. Therefore, after applying all mentioned techniques-

---

**TABLE 1. Module ablation experiment results**

| Resnet | MixUp | SE | DCN | mAP  | AP@.5 | AP@.75 | AP@S  | AP@M  | AP@L  | AR@L  | FPS |
|--------|-------|----|-----|------|-------|--------|-------|-------|-------|-------|-----|
| ✓      | 0.224 | 0.443 | 0.208 | 0.136 | 0.307 | 0.431 | 0.532 | 48    |
| ✓ ✓    | 0.236 | 0.449 | 0.215 | 0.131 | 0.326 | 0.462 | 0.549 | 59    |
| ✓ ✓ ✓  | 0.249 | 0.451 | 0.250 | 0.127 | 0.357 | 0.490 | 0.574 | 57    |
| ✓ ✓ ✓ ✓ | 0.252 | 0.461 | 0.242 | 0.135 | 0.335 | 0.476 | 0.563 | 47    |
| ✓ ✓ ✓   | 0.247 | 0.454 | 0.257 | 0.137 | 0.353 | 0.473 | 0.570 | 57    |
| ✓ ✓ ✓ ✓ | 0.272 | 0.474 | 0.269 | 0.164 | 0.385 | 0.609 | 0.664 | 52    |

**TABLE 2. Different feature maps with DCN**

| C3  | C4  | C5  | mAP  | AP@.5 | AP@.75 | AP@S  | AP@M  | AP@L  | AR@L  | FPS |
|-----|-----|-----|------|-------|--------|-------|-------|-------|-------|-----|
| ✓   | 0.272 | 0.474 | 0.269 | 0.164 | 0.385 | 0.609 | 0.664 | 52    |
| ✓   | 0.266 | 0.459 | 0.257 | 0.151 | 0.390 | 0.614 | 0.679 | 54    |
| ✓   | 0.262 | 0.462 | 0.251 | 0.146 | 0.387 | 0.625 | 0.682 | 54    |

**TABLE 3. The obtained results from different algorithms**

| Methods  | mAP  | AP@.5 | AP@.75 | AP@S  | AP@M  | AP@L  | AR@L  | FPS |
|----------|------|-------|--------|-------|-------|-------|-------|-----|
| SSD [6]  | 0.152 | 0.346 | 0.145 | 0.102 | 0.236 | 0.351 | 0.421 | 61  |
| FCOS [28]| 0.194 | 0.433 | 0.188 | 0.128 | 0.291 | 0.394 | 0.480 | 55  |
| YOLOv3 [10]| 0.224 | 0.443 | 0.208 | 0.136 | 0.307 | 0.462 | 0.549 | 59  |
| RetinaNet-101 [27]| 0.245 | 0.463 | 0.232 | 0.145 | 0.326 | 0.499 | 0.596 | 30  |
| YOLOv4 [11]| 0.251 | 0.469 | 0.265 | 0.159 | 0.352 | 0.505 | 0.572 | 56  |
| YOLOv5 [12]| 0.261 | 0.471 | 0.264 | 0.158 | 0.371 | 0.574 | 0.606 | 60  |
| Ours     | 0.272 | 0.474 | 0.269 | 0.164 | 0.385 | 0.609 | 0.664 | 52  |
the, mAP and AP@S improved by 4.8% and 2.8%, respectively, while the speed improved by 4 FPS.

To demonstrate that in the backbone network, the C3 feature map is responsible for detecting small objects, a deformable convolution is introduced in the C4 and C5 feature maps based on YOLOv3_Rvd_MU_SE and conducted experiments. As shown in Table 2, the obtained results in this regard show that as the network deepens, the effect of introducing deformable convolution gradually weakens. The AP@S of C4 and C5 dropped by 1.3% and 1.8%, respectively. It is concluded that in aerial images, the C3 feature layer is mainly responsible for small object detection.

E. COMPARATIVE EXPERIMENT

In order to evaluate the performance of the proposed method, all methods are studied under the same conditions. The obtained results in Table 3 indicate that the proposed method outperforms the baseline YOLOv3 from different aspects. More specifically, the receptive field of the deformable convolution is variable, and the features can be extracted more accurately. Compared with the one-stage detection algorithm RetinaNet-101 with ResNet-101 as the backbone network, it has more layers than the proposed algorithm. However, when extracting features, it does not aim at the problems of small objects and deformation in aerial images. Therefore, it is 2.7% lower than the proposed method and has a speed of 22 FPS. Compared with SSD, the proposed algorithm is 6 FPS behind, but it’s mAP is 12% higher than that of SSD. This may be attributed to the absence of a fully connected layer. Meanwhile, compared with the of FCOS, mAP is increased by 7.8%, and the speed is reduced by 3 FPS. Since FCOS is based on the center point for detection, it cannot achieve the desired effect in aerial scenes with a high concentration of objects. After comparing with classical detection methods, it is also compared with higher-order YOLOv4 and YOLOv5 algorithms. When compared with the YOLOv4 algorithm, the mAP of the proposed method is 2% higher than that of YOLOv4 and 1.1% higher than that of YOLOv5. However, deformable convolution, attention mechanism, and other technologies are embedded in the proposed method, so that detection speed decreased by 4 FPS and 8 FPS compared with YOLOv4 and YOLOv5, respectively. Thus, the tradeoff between performance and processing time against YoloV5 is not greater than 1%. In order to better reflect the difference between the proposed method and YoloV5, the precision and recall curves of each category are drawn in the test set.

As shown in Figure 13, due to the relatively regular shape of "car" and "bus", the area under the P-R curve of YOLOv5 is better than that of the proposed method. However, on objects that are prone to deformation such as "people" and "bike", the area under the P-R curve of the proposed method is better than that of YOLOv5. It was proven once again that the proposed method has better detection effect for small objects and deformed objects. Meanwhile, it provides an idea to improve advanced algorithms to adapt to aerial object detection.

F. RESULTS AND ANALYSIS

Figure 14 shows the comparison results between baseline and the proposed method in detecting small objects such as people. It is observed that 36 objects can be detected using YOLOv3, while 45 objects are detected using the proposed method, indicating an increment of 20%. It is worth mentioning that the sitting person that is different from ordinary people can also be detected through this method. In order to visualize the difference from YOLOv5 as shown in Figure 15, an aerial image with more categories is selected for prediction. It is observed that in the detection of "car" and "bus", the accuracy of YOLOv5 is higher than that of the proposed method. But for the detection of "people" and "bike" which are smaller and easily affected by deformation, the proposed method detects more objects than YOLOv5. It further verifies the better performance of the proposed method in aerial image.
V. CONCLUSION
In the present article, the aerial object detection problem is analyzed systematically and a detection method is proposed based on the YOLOv3. To this end, the deformable convolution and attention mechanism are applied to improve the detection accuracy. In this regard, the background information is enriched to improve the generalization ability by the MixUp method. Moreover, Soft-NMS is replaced with NMS to reduce the occlusion of objects in aerial scenes. Then experiments are carried out on the challenging VisDroneDet2018 dataset to evaluate the effectiveness of the proposed strategies. For deformable objects in aerial images, the proposed method is an effective scheme to improve other detection models. Although our work has achieved satisfactory results, there are still some issues that will be solved in a future work, such as recovering blurred images transmitted by UAVs and improving the detection ability of truncated objects. In the future, we need to find out the solution to solve these problems.

VI. ACKNOWLEDGEMENT
This research was supported by Education Department of Jilin Province of China (grant number JJKH20210570KJ) and the National Natural Science Foundation of China (grant number 62062064).

REFERENCES
[1] Z. Shao, G. Cheng, J. Ma, Z. Wang, J. Wang, and D. Li, “Real-time and accurate uav pedestrian detection for social distancing monitoring in covid-19 pandemic,” IEEE Transactions on Multimedia, 2021.
[2] G. Yan, Q. Sun, J. Huang, and Y. Chen, “Helmet detection based on deep learning and random forest on uav for power construction safety,” Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 25, no. 1, pp. 40–49, 2021.
[3] N. A. Khan, N. Hanchi, S. N. Brohi, R. S. A. Usmani, and A. Nayar, “Smart traffic monitoring system using unmanned aerial vehicles (uavs),” Computer Communications, vol. 157, pp. 434–443, 2020.
[4] N. K. Chauhan and K. Singh, “A review on conventional machine learning vs deep learning,” in 2018 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, 2018, pp. 347–352.
[5] J. Murphy, “An overview of convolutional neural network architectures for deep learning,” Microway Inc, 2016.
[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.
[7] Z. Li and F. Zhou, “Fsfd: feature fusion single shot multibox detector,” arXiv preprint arXiv:1712.00960, 2017.
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[9] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
[10] ——, “Yolo3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.
[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolo4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.
[12] G. Jocher, “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements,” https://github.com/ultralytics/yolov5, Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370
[13] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” arXiv preprint arXiv:1904.07850, 2019.
[14] X. Zhou, V. Koltun, and P. Krähenbühl, “Probabilistic two-stage detection,” arXiv preprint arXiv:2103.07461, 2021.
[15] L. Cui, R. Ma, P. Lv, X. Jiang, Z. Gao, B. Zhou, and M. Xu, “Mdsd: multi-scale deconvolutional single shot detector for small objects,” arXiv preprint arXiv:1805.07099, 2018.
[16] M. Kisantal, Z. Wojna, J. Murawski, J. Narumiec, and K. Cho, “Augmentation for small object detection,” arXiv preprint arXiv:1902.07296, 2019.
[17] 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 Sensing, vol. 12, no. 9, p. 1432, 2020.
[18] C. Deng, M. Wang, L. Liu, Y. Liu, and Y. Jiang, “Extended feature pyramid network for small object detection,” IEEE Transactions on Multimedia, 2021.
[19] C. Yang, Z. Huang, and N. Wang, “Querydet: Cascaded spare query for accelerating high-resolution small object detection,” arXiv preprint arXiv:2103.09136, 2021.
[20] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” IEEE access, vol. 7, pp. 128,637–128,669, 2019.
[21] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.
[22] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05). 1. IEEE, 2005, pp. 886–893.
[23] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971–987, 2002.
[24] T. Hastie, S. Rosset, J. Zhu, and R. Zou, “Multi-class adaboost,” Statistics and Its Interface, vol. 2, no. 3, pp. 349–360, 2009.
[25] T. Joachims, “Making large-scale svm learning practical,” Technical report, Tech. Rep., 1998.
[26] M. Pal, “Random forest classifier for remote sensing classification,” International journal of remote sensing, vol. 26, no. 1, pp. 217–222, 2005.
[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[28] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully convolutional one-stage object detection,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.
[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
[30] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.
[31] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.
[32] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
[33] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, pp. 91–99, 2015.
[34] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, “Towards open world object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5830–5840.
[35] S. Zhang, G. He, H.-B. Chen, N. Jing, and Q. Wang, “Scale adaptive proposal network for object detection in remote sensing images,” IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 6, pp. 864–868, 2019.
[36] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun, “Metaanchor: Learning to detect objects with customized anchors,” arXiv preprint arXiv:1807.00980, 2018.
[37] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, “Region proposal by guided anchoring,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2965–2974.
[38] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, “Rnet: A hybrid detector for object detection in drone-captured images,” in Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.
[39] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, “Libra r-cnn: Towards balanced learning for object detection,” in Proceedings of
the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 821–830.

[40] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 558–567.

[41] X. Zhu, H. Hu, S. Lin, and J. Dai, “Deformable convnets v2: More deformable, better results,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9308–9316.

[42] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[43] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-nms–improving object detection with one line of code,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 5561–5569.

[44] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu et al., “Visdrone-det2018: The vision meets drone object detection in image challenge results,” in Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.

[45] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” arXiv preprint arXiv:1710.09412, 2017.

***