Journal of Computer Science and Cybernetics, V.39, N.3 (2023), 293–314
DOI no 10.15625/1813-9663/18259

DATA AUGMENTATION ANALYSIS OF VEHICLE DETECTION IN AERIAL IMAGES

KHANG NGUYEN*

University of Information Technology, Vietnam National University, Ho Chi Minh City, Quarter 6, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Viet Nam

Abstract. Drones are increasingly used in various application domains, including surveillance, agriculture, delivery, search, and rescue missions. Object detection in aerial images (captured by drones) gradually gained more interest in the computer vision community. However, research activities are still very few in this area due to numerous challenges, such as top-view angles, small-scale objects, diverse directions, and data imbalance. In this paper, we investigate different data augmentation techniques. Furthermore, we propose combining data augmentation methods to further enhance the performance of state-of-the-art object detection methods. Extensive experiments on two public datasets, which are AERIAU and XDUAV, demonstrate that the combination of random cropped and vertical flipped data boosts the performance of object detectors on aerial images.

Keywords. Drone, object detection, vehicle detection, data augmentation.

1. INTRODUCTION

In computer vision, there are several fundamental visual recognition problems including image classification, object detection, instance segmentation, and semantic segmentation [1]. In particular, object detection performs two tasks: locating objects in images and assigning the corresponding class labels. Moreover, object detection is an essential part of many computer vision applications, for example, face recognition, object tracking, action recognition, and instance segmentation. In recent years, more research works based on deep learning have made a significant improvement in object detection [2–11].

With the progressive growth of Unmanned Aerial Vehicles (UAVs), automatically collected images in various altitudes and different angles have been gradually investigated in many computer vision research works. Aerial image analysis can be applied for surveillance and security monitoring, agriculture, delivery, and disaster management [12–15]. Their applications attract a great deal of attention, for example, object detection [16, 17], action recognition [18, 19], object tracking [20, 21], face recognition [22], and vehicle flow estimation [23, 24]. In particular, object detection is considered the most important task in aerial image analysis.

Aerial images are usually captured by UAVs such as drones or flycam with a bird's eye perspective. Additionally, the mobility of UAVs also creates more challenges compared to

*Corresponding author.
E-mail addresses: khangnttm@uit.edu.vn (K.Nguyen)

images in standard datasets such as ImageNet [25] or MS COCO [26] due to the following reasons:

- The scales of objects in aerial images are generally affected by the UAV's altitude. For example, when taken at lower altitudes, images might include a small number of objects with a high resolution; when UAVs fly to higher altitudes, more objects will be captured in a small size.
- The different spatial distribution of objects in the image, for example, crowded or sparse scenes.
- The mobility of a drone or flycam with a movable camera could capture different angles which leads to different visual appearances of the same object.
- The changes in weather and illumination (e.g., daytime, nighttime, sunny, cloudy, foggy, or rainy) also drastically affect the object's visibility and appearance.

In the past few years, the number of research works has increased but there are still some limitations that hinder their breakthrough in accuracy. The main reason came from the intrinsic characteristics of aerial images. Small-scale objects may lead to vanishing information through the process of resizing images before being put into the neural network. The data imbalance among classes is also challenging to build a model without data bias.

Recently, deep learning methods still encounter some limitations due to two reasons. Firstly, detectors are based on a pre-trained model for classification problems. Secondly, processing high-resolution aerial images is time-consuming and expensive. There are many efforts to overcome the challenges. For example, when training a deep learning model, the value of validation error and training error must be taken into consideration. The distributed deep learning servers aim to tackle the computational cost issue. Meanwhile, data augmentation is another efficient method expected to achieve the target mentioned above.

The novelty and contribution of the article are highlighted as follow:

- Firstly, in this article, the focus is on vehicle object detection. This is a challenging problem since vehicles have a wide range of directions, colors, and similarities between vehicles.
- Secondly, this study proposes combining different augmentation methods to improve object detection performance in aerial images.
- Last but not least, this study is the first one who conducts intensive experiments on large-scale datasets and performs various detection methods. Specifically, recent advanced deep models such as Faster RCNN [4], ATSS [27], GFL [28], YOLO [29], and FSAS [30] methods are investigated in this article. Other UAV-based detectors, which are D2Det [31], TPH-YOLOv5 [32], and DSHNet [33], are also utilized to explore the effects of data augmentation strategies

The rest of the paper is structured as follows. In Section 2, object detection methods, their variants for aerial images, and different data augmentation methods will be reviewed. In Section 3, different data augmentation methods will be detailed. In Section 4, the experiments and our discussion will be presented. Finally, some conclusions in will be in Section 5.

2. RELATED WORKS

2.1. Object detection methods

Object detection methods mainly focus on standard datasets such as PASCAL VOC [34], MS-COCO [35], and ImagetNet [25]. Fast R-CNN [36] has the following two stages, finding regions in the image that might contain an object; and then classifying objects in these regions. Region proposals are searched by using an algorithm such as Selective Search [37] or Edge Boxes [38]. Then, a classifier such as SVM is used to classify objects [39]. Later, Faster R-CNN [4] generates region proposals by using a region proposal network (RPN) which significantly speeds up the process. Mask R-CNN [5] is an extended version of Faster R-CNN which replaces RoI pooling with Align pooling so that Mask R-CNN can handle both object detection and instance segmentation. Despite being one-stage detectors, YOLO3 [29], SSD [40], and RetinaNet [9] are notable methods. One-stage methods only directly predict grid cells while skipping the generating region proposal step. This change in speed. However, for small-scale objects and class imbalance unlike PASCAL VOC [41], its performance reveals weaknesses. In that case, RetinaNet [9] was proposed as a one-stage object detection model that utilizes a focal loss function to address the above-mentioned problems during training.

As discussed in Section 1, object detection in aerial images is more challenging compared to images in standard datasets. To deal with various scales of objects which affected by different viewpoints, a network has been proposed to extract semantic features and refine spatial details of multi-scale objects in images [42]. [43] introduces the IOU-sampling method and a balanced ℓ_1 loss to alleviate the impact of imbalance in the dataset and increase the priority on inessential areas. Object detection methods in aerial images based on the ensemble of models such as HAL-RetinaNet [17], RetinaNets [9], DPNet [44], and Cascade R-CNN [45] have achieved some great results [46].

2.2. Data augmentation methods

Many methods approach data augmentation by using the cropping technique, for example, ClusDET [47] and DMNet [48] which focus on cropping images to small regions, and then detecting each individual one. The efficiency of the detector model depends on how the images are cropped, in other words, the less the scale of background in each region is, the better the model will perform. The object might be presented in different orientations. Previous research has suggested methods involving horizontal proposals but it is not suitable for aerial images. RoI Transformer [49] was proposed to address this problem.

Therefore, in this paper, this study focuses on small objects, omnidirectional objects, and size diversity in the same object by applying data augmentation techniques.

3. DATA AUGMENTATION ANALYSIS

3.1. Public datasets for training

In this subsection, first, the challenges occurring in aerial images will be discussed. Then, the datasets used in the benchmark suite will be briefly introduced.

3.1.1. Dataset challenges

Most object detection models encounter difficulties and perform poorly with aerial dataset due to the following attributes:

- **Size diversity**: Images taken by UAVs are usually taken at 10 meters or up to 5 kilometers, consequently creating a lot of different sizes of the same object.
- **Different perspective**: Aerial images are largely taken from above, while most conventional datasets are taken at a lower altitude which results in some models being unable to recognize the already-learned classes.
- **Small object**: Objects in aerial images usually have only a few dozen pixels or only a few, even in high-resolution images, which makes any information or feature extracted very little.
- **Omnidirectional**: There is no limit to how objects are posed in aerial images because they are similar to the bird's-eye view, unlike conventional datasets where, pedestrians, for example, are always upright.
- **Complex image**: Aerial images have an extremely wide view and tend to include a large proportion of background where many objects that can be easily confused with objects that need to be predicted.

For the aforementioned reasons, it is challenging to train a good model for object detection in aerial images, especially when there is a lack of data. In the following subsections, a summary of two public datasets used to create a non-photographic dataset for training our object detector will be presented (Figure 1).

3.1.2. AERIAU dataset

AERIal AUgmentation dataset (AERIAU) Dataset [50] consists of 1,474 training images and 184 testing images. The images are captured in a top-down viewpoint with different ratios. There are four categories, namely, "car", "truck", "bus", and "motor".

3.1.3. XDUAV dataset

XDUAV Dataset [51] was collected by using DJI Phantom 2 at an average altitude of 100m in Xi'an, China's rural and urban areas. The dataset consists of 11 videos which are recorded in resolution 1920 x 1080, 30fps. There are 4,344 images, 3,475 training images, and 869 testing images. The categories include "car", "bus", "truck", "tanker", "motor", and "bicycle".

3.2. Data augmentation strategies

This section describes clearly data augmentation based on basic geometry transformations including Rotation, Crop, Flip, and Zoom (Figure 2). For the implementation, I adopted the PIL library developed by Lundh *et al.* [52].

Figure 1: Exemplary images in AERIAU (top row) and XDUAV (bottom row) datasets

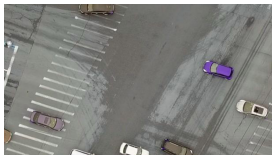
(a) Input Image

(b) Random Rotation

(c) Erasing

(d) Horizontal Flip

(e) Vertical Flip



(f) Zoom

(g) Brightness

(h) Random Crop

Figure 2: Single data augmentations: (a) Input Image, (b) Random Rotation, (c) Random Erasing, (d) Random Horizontal Flip, (e) Random Vertical Flip, (f) Random Zoom, (g) Random Brightness, (h) Random Crop.

3.2.1. Single data augmentation

- **Random Rotation**: This is done by rotating the image at a random θ degree in the range of [10, 180] around the center of the image. In a straight-down view, rotation enhances the diversity of the subject. After rotating theta degrees, the coordinate of bounding boxes is updated by Equation 1

$$\begin{cases} x' = (x - C_x) \times \cos(\theta) - (y - C_y) \times \sin(\theta) + C_x \\ y' = (x - C_x) \times \sin(\theta) - (y - C_y) \times \cos(\theta) + C_y, \end{cases} \tag{1}$$

where, x', y': new coordinate of bounding boxes; x, y: original coordinate of bounding boxes; θ: rotation angle; C_x, C_y: center coordinate of the image.

- **Random Crop**: is the processing step to fix the image size. Each image will cut out four random non-intersecting regions, each image is 500x225 in size. The images that are passed through the network architecture will be resized in smaller sizes regularly. Small objects in the image lose information. After cropping, images will reduce the information of small objects when passed through the network architecture.

  After randomly cropping, I received four parameters x_c, y_c, w_c, h_c which are coordinates of two top-left, width, and height of the cropped region. Then, with each bounding box inside the cropped region, I update the annotation by Equation 2

$$\begin{cases} x'_{\min} = \max(x_{\min} - x_c, 0) \\ y'_{\min} = \max(y_{\min} - y_c, 0) \\ x'_{\max} = \min(x_{\max} - w_c - 1, w_c) \\ y'_{\max} = \min(y_{\max} - h_c - 1, h_c). \end{cases} \tag{2}$$

  However, some bounding boxes may be outside the cropped area, so I will remove it, the removal condition is as Equation 4

$$\begin{cases} x'_{\min} > w_c \\ y'_{\min} > h_c \\ x'_{\max} < 0 \\ y'_{\max} < 0. \end{cases} \tag{3}$$

- **Random horizontal/vertical flip**: also referred to horizontal/vertical flipping, is much more common than vertical flipping. Two functions RandomHorizontalFlip and RandomVerticalFlip take an input of 1 image PIL with probability p for the image to be flipped, the default parameter value of p is 0.5, and the value range is 0 to 1. The output image is the same size as the input image, the image can be flipped or not depending on the probability p when calling the function. Specific processes will be represented in Pseudocode 1.

Algorithm 1: Random Horizontal/Vertical Flip

Data: Image I and its list of bounding boxes B
Result: New flipped image I and new list of boxes L_B

```
Define empty new bounding boxes list L_B.
p ← [0,1]
if p ≥ 0.5 then
    Image I* is horizontally/vertically flipped from image I.
    I* ← I
    for each b in B do
        temp_{y_min}, temp_{y_max} ← y_min, y_max
        y_min ← h − temp_{y_max}
        y_max ← h − temp_{y_min}
        Add new coordinate (x_min, y_min, x_max, y_max) to L_B
    end
end
```

(a) Image-aware Random Erasing (b) Object-aware Random Erasing (c) Image and Object-aware Random Erasing

Figure 3: Examples of three random erasing methods

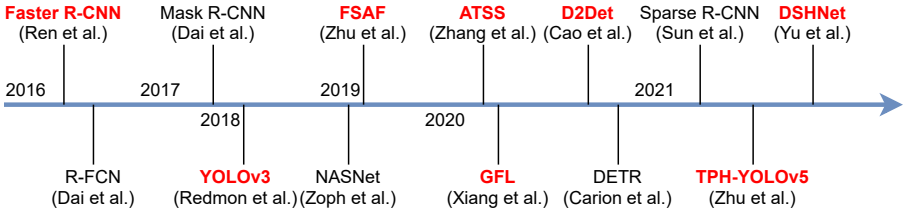


Figure 4: Timeline of state-of-the-art object detection methods. The benchmarked methods are marked in red and boldfaced font.

- **Random zoom**: Each image will create its zoom version, with the same size at a random position in the image. After zooming the image, I update the coordinates of the bounding boxes according to Equation 4

$$\begin{cases} x' = \dfrac{w}{2} + \dfrac{1}{R} \times (x - \dfrac{w}{2}) \\ y' = \dfrac{h}{2} + \dfrac{1}{R} \times (y - \dfrac{h}{2}), \end{cases} \tag{4}$$

  where, x', y': new coordinate of bounding boxes; x, y: original coordinate of bounding boxes; R: zoom ratio in a range of [0.1, 1]; w, h: width and height of bounding boxes.

- **Random brightness**: Images can be augmented by random darkening or brightening. Values change in paragraph [0.5, 1.5]. The brightness is kept the same at the value of 1. The higher the value is, the brighter the image becomes, and vice versa.

- **Random erasing**: was proposed by Zhong *et al.*[53]. Here, occlusion is considered an important factor in image processing. Therefore, Random Erasing is applied for data augmentation. Random Erasing randomly selects a rectangular area in the image area, bounding box, and replaces the pixels in this area with a random value. There are three erasing methods: The image, the bounding box, and both. The examples of three methods are shown in Figure 3.

3.2.2. Combined data augmentation

As aforementioned, object detection in aerial images has many challenges due to many reasons, i.e., small objects, erratic distribution, object imbalances between classes as well as between object and background, and various camera angles. Therefore, building a general deep learning model which yields good results on the testing set (unseen) may face many problems. For example, models that have an infertile generalization encounter overfitting on the training set. Therefore, data augmentation is an extremely effective method [54] to tackle this overfitting problem.

It is observed that objects in the aerial image can appear in any direction with various camera angles. Therefore, considering objects at various angles makes the dataset more robust. Some methods of augmenting data that are usually used in the training step: Translation, flip, and rotation [55].

Objects of interest usually occupy small regions in high-resolution images. Therefore, when put into the training step, it is necessary to reduce the size of the image to suit the network architecture, which causes a lot of information loss for small objects. To retain the information about the object, training the object in multiple ratios, diverse truncate objects, image cropping, and zooming techniques are often applied [46, 56].

Data augmentation techniques are applied to help diversify data and build a general model. In this paper, different data augmentation combinations in training object detectors will be investigated.

3.3. Object detection methods

Object detectors are required for benchmarking purposes. Figure 4 depicts the timeline of different methods used in this paper. The details of these methods are presented in the following subsections.

3.3.1. One-stage detection methods

You Only Look Once – YOLOv3 [29]. Different from the previous methods YOLO v1 and v2, YOLOv3 softmax with independent logistic regression for multi-label classification. Besides that, YOLOv3 used Feature Pyramid Networks – FPN and returned three predictions at different scale positions of the feature map. This made YOLOv3 take advantage of various complexity of feature maps for predictions. The Darknet-53 architecture employed many residual blocks like ResNet, YOLOv3 used this as a feature extractor for Darknet-19.

Feature Selective Anchor-Free - FSAF [30]. Zhu *et al.* proposed a basic block that can be attached with single-shot object detectors. In the training process, FSAF automatically points to each object by its suitable feature level. In the interpolating process, FSAF can compact with parallelly solved anchor-based branches. The general concept of the FSAF module is online feature selection applied to the training of multi-level anchor-free branches. During the training process, FSAF dynamically assigns each instance to the most suitable feature level.

Adaptive Training Sample Selection - ATSS [27] is the method of object detection, which automatically chooses positive and negative samples based on its statistical feature. This method has improved significantly the performance. Moreover, it also narrows the gap

Table 1: The number of objects before and after augmentation on the AERIAU Dataset [50].

Dataset	Car	Truck	Bus	Motor	Total
Original	52,797	3,099	1,666	5,623	63,185
Random Rotation	+52,797	+3,099	+1,666	+5,623	+63,185
Random Crop	+28,172	+1,553	+12,757	+41,482	+83,964
Random HorizontalFlip	+52,797	+3,099	+1,666	+5,623	+63,185
Random VerticalFlip	+52,797	+3,099	+1,666	+5,623	+63,185
Random Zoom	+25,441	+1,553	+1,102	+2,947	+31,043
Random Brightness	+422,376	+26,225	+13.328	+44,984	+505,480
Random Erasing	+87,745	+5,265	+2,793	+9,744	+105,547

between the anchor-based detector and anchor-free detector because, in essence, the basic difference between these two detectors is the selection of positive and negative samples for the training step.

3.3.2. Two-stage detection methods

Faster RCNN [4]. Proposal regions in the Faster-RCNN method [34] are based on the RPN Network, which is a fully convolutional network that receives an image with any size as the input and returns a set of object proposal regions in each position of the feature map. Each position is represented by a feature vector, which will be passed through the classifier (object or no object) and bounding box regression layer. These will go through the last layer for object classification and positioning.

Generalized Focal Loss - GFL [28] is the method towards the unification regarding the calculation of the training and testing process and enhances the performance and accuracy from these. Regularly, classification scores and IOU centeredness scores are independent, which is not optimized. Therefore, the GFL method proposed a solution for compacting three methods, *i.e.*, Focal Loss, Quality Focal Loss, and Distribution Focal Loss.

3.3.3. UAV-based detectors

Many studies proposed specific detectors for aerial images captured from UAVs and drones. For better comparisons, this study uses three benchmark UAV-based detectors to analyse the effectiveness of data augmentation methods, which are D2Det, DSHNet, and TPH-Yolov5. In this section, an overview of the main contributions of these three methods is presented.

D2Det [31] is a two-stage detector proposed to improve the localization and classification task. To enhance the performance of bounding box regression, Cao *et al.* [31] proposed the dense local regression module, which was improved by a binary overlap prediction strategy that helped reduce the influence of background region on the final box regression. Besides, the performance of the classification task was also pushed by a proposed discriminative RoI Pooling scheme, which sampled from various sub-regions of the proposal and performed adaptive weighting to achieve discriminative features.

DSHNet [33] was proposed to handle the class imbalance problem in object detection in aerial images, which is one of the main challenges in images captured from drones. This

challenge commonly leads to poor performance in tail classes. The key modules in the DSHNet detector contained Class-Biased Samplers (CBS) and Bilateral Box Heads (BBH). The CBS is the sampler, which was used to increase the number of head and tail classe samples. The BBH module contained two branch classifiers; each branch was responsible for classifying corresponding classes (tail or head classes).

TPH-YOLOv5 [32] is the YOLOv5-based detector, which is a one-stage method. Overall, TPH-YOLOv5 was proposed to solve several challenges: different attitudes, various scales, motion blur, and dense objects. In detail, based on YOLOv5, Zhu *et al.* [32] added one more prediction head to detect various-scale objects. Furthermore, they proposed to replace the original prediction heads with Transformer Prediction Heads (TPH) to explore the effect of the self-attention mechanism on the prediction potential. Besides, the authors aggregated the convolutional block attention model (CBAM) to find attention on situations with dense objects. Finally, they provided useful pipelines for training and inference, such as data augmentation techniques and multi-scale inference.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Experimental settings

To evaluate data augmentation techniques, I selected AERIAU and XDUAV as two benchmarks for object detection in UAV images. As mentioned earlier, I applied five object detection methods in the MMDetection toolbox [57] including Faster R-CNN, ATSS, GFL, YOLOv3, and FSAS, where Faster R-CNN, ATSS, and GFL represent two-stage detectors; and YOLOv3, and FSAS represent one-stage detectors. Our experiments were run on 2 GPUs RTX2080 TI, RAM 24GB. I selected ResNet50 as the backbone architecture for all methods and used its pre-trained models (including FPN). Results are evaluated based on MS COCO API. In particular, I adopt AP, AP_{50}, and AP_{75} as metrics to measure the precision, where AP_{50} and AP_{75} are values of AP at the IoU threshold of 0.5 and 0.75, respectively. Finally, the mean AP value (mAP) is computed across all categories.

4.2. Experimental results on AERIAU

The first experiment was executed on the AERIAU dataset, which was captured from a straight camera angle. In other words, this experiment evaluated the impact of single and multiple augmentation techniques for straight viewing angles. The data augmentation techniques were applied to the training set. The number of objects after augmentation is shown in Table 1. Meanwhile, Table 2 shows the results of different data augmentation methods over various object detectors. As seen in the table, object detection methods yield different results on the given data augmentation approaches. ATSS achieves the best performance in AERIAU (original), vertical flipping, random zoom, and random erasing. In the meantime, GFL reaches the top spot in random rotation and random cropping. Faster RCNN obtains the best results in random brightness and horizontal flipping. The average result for each data augmentation method was further computed. As shown in Table 2, the RandomCrop technique shows the best result for all three metrics (AP, AP_{50}, and AP_{75}). While applying RandomCrop, more images will be generated from random regions on the original image, this helps the object detector learn on smaller images instead of the complicated original

Table 2: Experimental results with **single data augmentation** strategies on the AERIAU dataset [50]. The mean results are highlighted in boldfaced font, and the top-2 mean results are marked in red and blue. Please view this table in the color pdf.

Method	**mAP**	**mAP$_{50}$**	**mAP$_{75}$**
Data Augmentation: AERIAU			
Faster RCNN	41.50	59.50	51.10
ATSS	45.70	64.70	56.10
GFL	43.10	62.50	53.80
YOLO	40.40	40.36	28.82
FSAS	37.80	59.20	45.70
Mean	**41.70**	**57.25**	**47.10**
Data Augmentation: AERIAU + Random Rotation			
Faster RCNN	32.60	49.20	40.20
ATSS	36.70	54.30	44.20
GFL	47.00	69.50	59.20
YOLO	39.80	39.84	7.48
FSAS	27.70	46.10	32.30
Mean	**36.76**	**51.79**	**36.68**
Data Augmentation: AERIAU + Random Crop			
Faster RCNN	48.40	68.60	60.70
ATSS	40.90	59.50	51.50
GFL	51.70	74.00	65.10
YOLO	26.90	26.86	19.46
FSAS	43.00	62.90	54.10
Mean	**42.18**	**58.37**	**50.17**
Data Augmentation: AERIAU + HorizontalFlip			
Faster RCNN	43.60	62.40	52.30
ATSS	43.40	63.10	54.40
GFL	43.30	62.60	54.60
YOLO	39.00	39.03	30.08
FSAS	27.30	44.40	33.20
Mean	**39.32**	**54.31**	**44.92**
Data Augmentation: AERIAU + VerticalFlip			
Faster RCNN	41.80	60.00	50.00
ATSS	42.40	60.00	52.70
GFL	41.20	60.00	51.60
YOLO	39.40	39.40	26.43
FSAS	38.10	58.60	50.20
Mean	**40.58**	**55.60**	**46.19**
Data Augmentation: AERIAU + Random Zoom			
Faster RCNN	44.20	62.50	55.20
ATSS	47.60	67.90	59.00
GFL	41.20	59.90	50.60
YOLO	32.40	32.38	23.22
FSAS	44.20	64.70	56.30
Mean	**41.92**	**57.48**	**48.86**
Data Augmentation: AERIAU + Random Brightness			
Faster RCNN	46.30	66.90	57.00
ATSS	33.70	48.10	41.80
GFL	40.90	59.20	51.20
YOLO	30.50	30.47	20.16
FSAS	33.50	53.40	42.60
Mean	**36.98**	**51.61**	**42.55**
Data Augmentation: AERIAU + Random Erasing			
Faster RCNN	33.10	49.50	41.00
ATSS	44.30	63.50	57.30
GFL	40.80	58.30	52.90
YOLO	29.30	29.32	20.87
FSAS	29.50	47.60	36.10
Mean	**35.40**	**49.64**	**41.63**

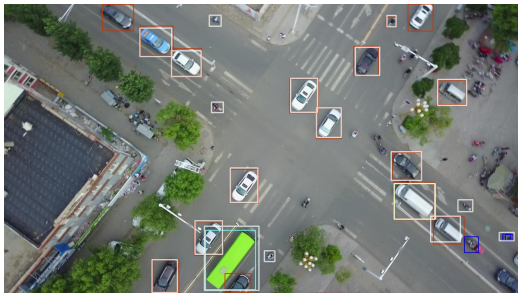
Table 3: Experimental results with **combined data augmentation** strategies on the AERIAU dataset [50]. The mean results are highlighted in boldfaced font, and the top-2 mean results are highlighted in red and blue. Please view this table in the color pdf.

Method	**mAP**	**mAP$_{50}$**	**mAP$_{75}$**
AERIAU + Random Crop + Random Zoom			
Faster RCNN	34.90	49.90	42.40
ATSS	45.30	64.90	56.10
GFL	49.60	69.70	60.80
YOLO	34.60	34.55	23.05
FSAS	43.60	62.30	53.90
Mean	**41.60**	**56.27**	**47.25**
AERIAU + Random Crop + Vertical Flip			
Faster RCNN	45.80	64.00	54.80
ATSS	49.50	69.60	62.10
GFL	47.20	68.10	57.50
YOLO	35.30	35.28	23.05
FSAS	46.10	66.10	58.30
Mean	**44.78**	**60.62**	**51.15**
AERIAU + Random Zoom + Vertical Flip			
Faster RCNN	37.20	53.10	45.60
ATSS	48.40	67.30	61.90
GFL	46.60	66.90	58.00
YOLO	44.40	44.44	30.82
FSAS	44.70	62.70	55.50
Mean	**44.26**	**58.89**	**50.36**
AERIAU + Random Crop + Horizontal Flip			
Faster RCNN	41.30	60.20	50.50
ATSS	46.80	65.80	58.30
GFL	48.90	69.00	61.80
YOLO	30.50	30.52	22.45
FSAS	43.60	62.90	55.20
Mean	**42.22**	**57.68**	**49.65**
AERIAU + Random Crop + Random Zoom + Vertical Flip			
Faster RCNN	40.10	56.90	47.90
ATSS	48.20	65.60	61.40
GFL	48.70	68.90	61.40
YOLO	39.60	39.63	26.30
FSAS	46.20	65.00	56.90
Mean	**44.56**	**59.21**	**50.78**

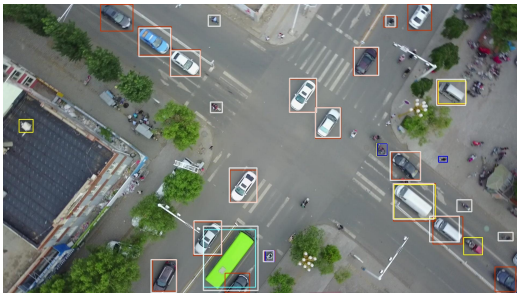
image. This method also reduces the background complexity. The runner-up is the Zoom technique, objects in images will be zoomed in when applying this technique, which helps to solve the challenge of tiny objects. Meanwhile, Random Vertical Flip, which enhances the dataset by rotating the image vertically, shows the third-highest result. The random data augmentation techniques show a positive effect in aerial images, especially on tiny objects in the image.

The results of different data augmentation techniques visualized in Figure 5. RandomCrop showed good effectiveness on objects belonging to bicycles, motors, and tiny objects at the corner. Since some objects may be cropped all, bounding boxes for them need to be updated. This makes the trained model extremely sensitive to tiny objects, for example, entering or leaving the scene (as shown in Figure 5c).

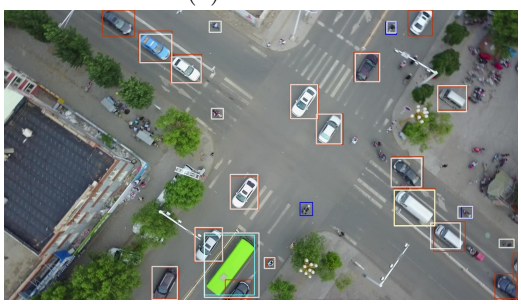
Taking advantage of each individual data augmentation method, different augmentation



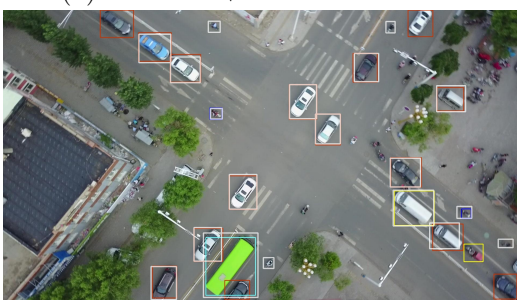
(a) AERIAU



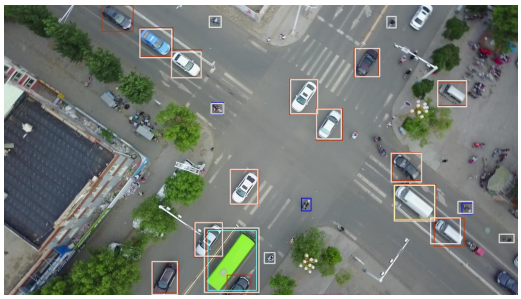
(b) AERIAU + Random Rotation



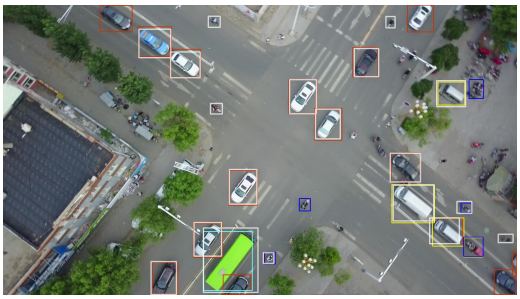
(c) AERIAU + Random Crop



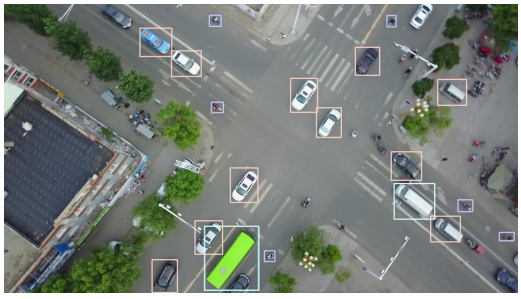
(d) AERIAU + Random HorizontalFlip



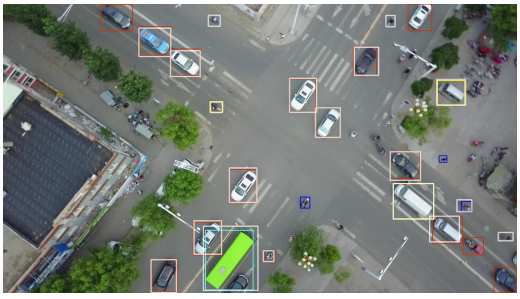
(e) AERIAU + Random VerticalFlip



(f) AERIAU + Random Zoom

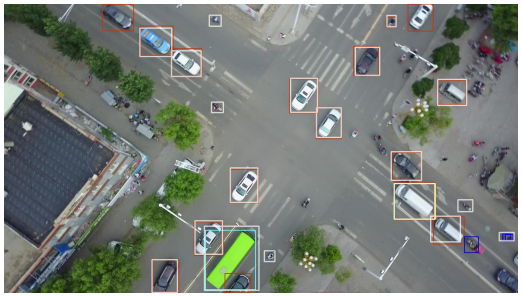


(g) AERIAU + Random Brightness

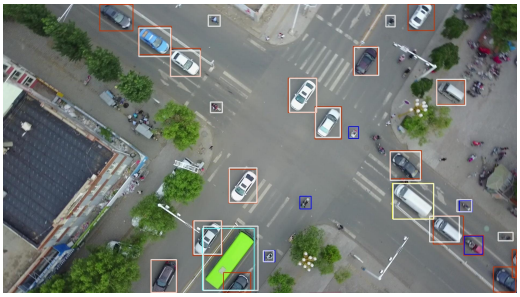


(h) AERIAU + Random Erasing

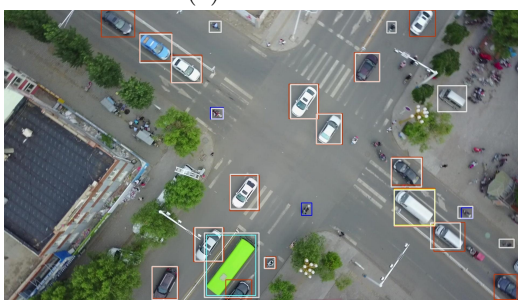
Figure 5: Faster RCNN: Visualization results of Single Data Augmentation Strategies. Color legend: truck, car, bus, motor, and Ground Truth.



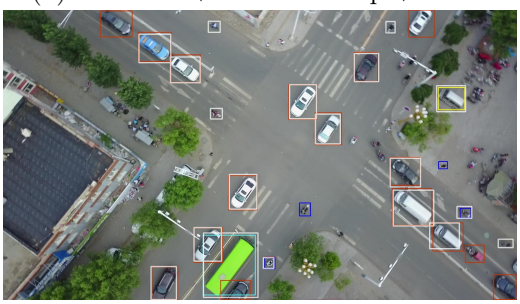
(a) AERIAU



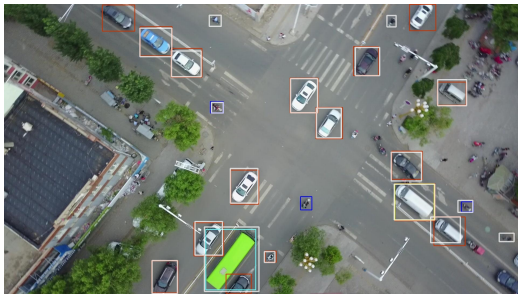
(b) AERIAU + Random Crop + Zoom



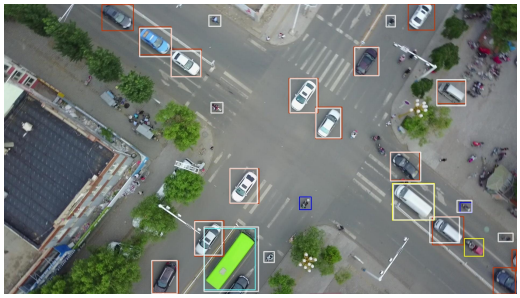
(c) AERIAU + Random Crop + Vertical Flip



(d) AERIAU + Random Zoom + Vertical Flip



(e) AERIAU + Random Crop + Horizontal Flip



(f) AERIAU + Random Crop + Zoom + Vertical Flip

Figure 6: Faster RCNN: Visualization results of Combined Data Augmentation Strategies. Color legend: truck, car, bus, motor, and Ground Truth.

methods were combined further and the results are shown in Table 3. Regarding object detection methods, ATSS obtains the best performance in the combinations of Random Crop + Random Zoom, Random Crop + Vertical Flip, and Random Zoom + Vertical Flip. Meanwhile, GFL reaches the top results in Random Crop + Horizontal Flip, and Random Crop + Random Zoom + Vertical Flip. Regarding data augmentation methods, Random Crop and Vertical Flip achieve the top-2 results. Due to aerial image capture, vehicles may appear in different directions. However, the view of images usually depends on the intent of the drone controller, the device that displays images is a smartphone (horizontal). In other words, vehicles usually move from left to right of the screen and vice versa. Therefore, the Vertical Flip technique helps increase images following the flow of movement. The

Table 4: The number of objects before and after data augmentation on the XDUAV Dataset [51].

Data Augmentation	Car	Truck	Bus	Motor	Bicycle	Tanker	Total
Original	26,885	2,275	2,140	5,278	1,598	138	38,314
Random Crop	+7,580	+879	+528	+1,124	+418	+73	+10,602
Random Vertical Flip	+26,885	+2,275	+2,140	+5,278	+1,598	+138	+38,314

combination of Zoom and Random Crop achieves the second-highest result among the rest of the combinations. The third-highest result is the combination of Zoom and Vertical Flip.

Figure 6 indicates that the combination of Random Crop and Zoom is sensitive to object "motor" while the Random Crop -Vertical Flip combination shows better effectiveness on "pedestrian". However, the dataset includes some unlabeled tiny objects leads to confusion in the training process. Therefore, the trained model might not recognize objects included in the image. The attempt to combine all these three methods makes the trained model detect several objects that appear in a small part of the image, but these objects are unlabeled. Thus this combination achieves the second-highest result.

4.3. Experimental results on XDUAV

Following the evaluation of the AERIAU dataset, another set of experiments on XDUAV including 6 vehicle categories: "car", "bus", "truck", "tanker", "motor", and "bicycle" were conducted. After applying the Random Crop and Vertical Flip techniques on the train, the number of objects obtained is shown in Table 4. Single and hybrid data augmentation techniques consisting of Random Crop, Random Vertical Flip, and Random Crop combined with Random Vertical Flip were applied.

I adopt two baselines, namely, Faster R-CNN and Faster R-CNN using FPN, as recommended in [58]. As shown in Table 5, Faster R-CNN with FPN yields better AP, AP@50, and AP@75 results compared with Faster R-CNN. In particular, it gains 4.2%, 2%, and 5.6%, in terms of AP, AP@50, and AP@75, respectively. I conducted another experiment using Random Crop, which just got bounding boxes of useful objects (objects have almost full bounding boxes after being cropped), this attempt helped significantly improve and achieved the highest result on "car". Meanwhile, the combination of Random Crop and Vertical Flip shows the highest result on all three metrics AP, AP@50, and AP@75. Especially, as seen in Figure 7, this combination yields the highest result on "bicycle", which is regarded as the most challenging object in the dataset.

To strengthen the effectiveness of the combined augmentation strategy, the experiments on three UAV-based detectors including D2Det, TPH-YOLOv5, and DSHNet, which are explicitly designed for object detection in aerial images were extended, or the experiments on the UAV dataset have been conducted by its authors. The results are reported in Table 6 and Table 7.

The results of the AP metric for each class are shown in Table 6. Obviously, the detector trained with an augmented dataset shows the performance improvement compared to its counterparts trained without augmenting data. In which, D2Det detector trained with ResNet-50 backbone performs the best among others on two vehicle categories: "truck" (AP 82.9%) and "bus" (AP 84.1%) while the DSHNet trained with ResNet-101 shows the

Table 5: Detection Results % of different object detection methods, i.e., Faster RCNN and Faster RCNN with FPN, and different data augmentation strategies on the XDUAV Dataset [51]. The top-3 results are highlighted in red, blue, and green, respectively. Please view this table in the color pdf.

Method	Car	Truck	Bus	Motor	Bicycle	Tanker	mAP	mAP_{50}	mAP_{75}
Faster RCNN	79.50	80.00	79.00	48.80	34.90	72.90	65.80	91.00	76.10
Faster RCNN + RandomCrop	79.40	80.90	80.30	48.90	34.30	73.10	66.10	91.00	77.00
Faster RCNN + VerticalFlip	80.20	81.20	80.60	51.00	38.10	73.10	67.40	91.70	78.40
Faster RCNN + RandomCrop + VerticalFlip	80.30	82.00	81.30	50.40	37.50	75.30	68.00	92.30	78.10
Faster RCNN + FPN	80.70	81.80	83.00	54.70	46.60	71.90	70.00	93.00	81.70
Faster RCNN + FPN + RandomCrop	81.10	82.00	83.20	54.40	48.40	72.70	70.30	92.70	82.00
Faster RCNN + FPN + VerticalFlip	80.70	83.10	83.90	54.60	48.60	71.80	70.40	93.40	82.70
Faster RCNN + FPN + RandomCrop + VerticalFlip	80.80	82.90	84.20	54.40	48.70	74.00	70.80	93.70	82.70

Table 6: Detection Results % AP of each vehicle category of three UAV-based object detection methods, i.e., D2Det, TPH-YOLOv5, DSHNet, and the combined augmentation method on the XDUAV Dataset [51]. The top-3 results are highlighted in red, blue, and green, respectively. Please view this table in the color pdf.

Method	Backbone	Image size	Car	Truck	Bus	Motor	Bicycle	Tanker
D2Det	ResNet50	1333 × 800	81.8	81.2	83.2	54.6	47.3	75.1
D2Det	ResNet101	1333 × 800	81.3	81.4	83	54	47.7	71.7
TPH YOLOv5	CSPDarknet53	800 × 800	77.2	79.2	80.3	54.4	37.9	12.9
DSHNet	ResNet50	1333 × 800	77.6	75.3	76.8	54.1	43.5	64.4
DSHNet	ResNet101	1333 × 800	78.3	76.8	78.9	55.1	47.3	72.1
D2Det + RandomCrop + VerticalFlip	ResNet50	1333 × 800	81.5	82.9	84.1	56.1	50	70.8
D2Det + RandomCrop + VerticalFlip	ResNet101	1333 × 800	81	82	83.9	54.8	49.8	75.3
TPH YOLOv5 + RandomCrop + VerticalFlip	CSPDarknet53	800 × 800	78.6	71.8	83.4	53.1	27.4	58
DSHNet + RandomCrop + VerticalFlip	ResNet50	1333 × 800	79.4	79.1	81.6	56.6	50.8	76.3
DSHNet + RandomCrop + VerticalFlip	ResNet101	1333 × 800	79.6	80	82.6	57.6	50.9	77

Table 7: Detection Results % on AP, AP@50 and AP@75 of three UAV-based object detection methods, i.e., D2Det, TPH-YOLOv5, DSHNet, and the combined augmentation method on the XDUAV Dataset [51]. The top-3 results are highlighted in red, blue, green, respectively. Please view this table in the color pdf.

Method	Backbone	Image size	AP	AP@50	AP@75
D2Det	ResNet50	1333 × 800	70.5	93.4	82.8
D2Det	ResNet101	1333 × 800	69.9	93.1	82.5
TPH YOLOv5	CSPDarknet53	800 × 800	57	80.9	65.9
DSHNet	ResNet50	1333 × 800	65.3	94.5	77.1
DSHNet	ResNet101	1333 × 800	68.1	95.8	78.8
D2Det + RandomCrop + VerticalFlip	ResNet50	1333 × 800	70.9	93.8	83.1
D2Det + RandomCrop + VerticalFlip	ResNet101	1333 × 800	71.1	94	84
TPH YOLOv5 + RandomCrop + VerticalFlip	CSPDarknet53	800 × 800	62.1	84.7	71.9
DSHNet + RandomCrop + VerticalFlip	ResNet50	1333 × 800	70.6	97.2	82.1
DSHNet + RandomCrop + VerticalFlip	ResNet101	1333 × 800	71.3	97.3	82.7

(a) Faster RCNN

(b) Faster RCNN + FPN

(c) Faster RCNN + RandomCrop

(d) Faster RCNN + FPN+ RandomCrop

(e) Faster RCNN + VerticalFlip

(f) Faster RCNN + FPN+ VerticalFlip

(g) Faster RCNN + RandomCrop+ VerticalFlip

(h) Faster RCNN + FPN+ RandomCrop + VerticalFlip

Figure 7: Visualization results of Data Augmentation Strategies on XDUAV Dataset. Color legend: car and motor.

best performance on "motor" (AP 57.6%), "bicycle" (AP 50.9%), and "tanker" (AP 77%). Furthermore, the DSHNet detector trained with augmented data shows a significant improvement on "tanker", whose AP is 4.9% higher than the original DSHNet. Training TPH-YOLOv5 with augmented data also performs better on "bus", whose AP is 3.1% higher than the original TPH-YOLOv5. The performance on other object categories of three detectors, D2Det, TPH-YOLOv5, and DSHNet, can be improved approximately from 0.9% to 2.7%.

The results of AP, AP@50, and AP@75 metrics are shown in Table 7. Undoubtedly, all top-2 results of the three detectors are achieved via training with augmented data. In which, DSHNet trained with ResNet-101 achieves the best results on AP and AP@50 metrics, which are 71.3% and 97.3%, respectively, while D2Det trained with ResNet-101 achieves the best result on AP@75, which is 84%. Besides, the AP and AP@50 scores of D2Det and DSHNet are competitive; the differences are only 0.2% and 0.1%, respectively. Furthermore, the results also show that the detector's performance with augmented data is improved with their counterparts trained with the original dataset. With D2Det using ResNet-101 backbone trained with augmented data, the AP and AP@50 are both 0.6% higher than the D2Det using ResNet-50 trained with original data while the AP@75 is 1.2% higher (I only compare the top results of the two backbones in each method. The D2Det trained with ResNet-101 performs better in standard training, but is reversed on the D2Det detector trained with augmented data). TPH-YOLOv5 trained with augmented data has a significant improvement, whose AP, AP@50, and AP@75 are 5.1%, 3.8%, and 6% higher than its counterpart. The DSHNet detector trained with augmented data also performs better than its counterpart when the AP, AP@50 and AP@75 scores are 3.1%, 1.5%, and 3.9% higher, respectively.

5. CONCLUSION AND FUTURE WORK

In conclusion, this study investigates the impact of data augmentation over object detectors in aerial images. This problem indeed is challenging due to some specific challenges caused by UAVs themselves. Our experimental results demonstrate the importance of data augmentation towards object detectors. This contributes to the performance enhancement of detecting objects. The results also indicate that Random Crop, Random Vertical Flip and Random Zoom are beneficial to training object detectors for aerial images.

Still, object detection methods on UAV images encounter many challenges. In the future, the data that impacts on training and improves the effectiveness of the state-of-the-art models will be explored. The issue of detecting tiny objects such as "motorcycle" or "bicycle" in a tightly packed area is also further investigated.

ACKNOWLEDGMENT

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number B2023-26-01.

REFERENCES

[1] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[3] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[6] 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.

[7] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[8] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[9] 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.

[10] B. Singh, M. Najibi, and L. S. Davis, "Sniper: Efficient multi-scale training," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018, pp. 9310–9320. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/166cee72e93a992007a89b39eb29628b-Paper.pdf.

[11] K.-D. Nguyen, K. Nguyen, D.-D. Le, D. A. Duong, and T. V. Nguyen, "Yada: You always dream again for better object detection," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 28 189–28 208, 2019.

[12] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, vol. 2, 2009, pp. 82–85.

[13] P. K. R. Maddikunta *et al.*, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors Journal*, 2021.

[14] M. Perreault and K. Behdinan, "Delivery drone driving cycle," *IEEE Transactions on Vehicular Technology*, 2021.

[15] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[16] G.-S. Xia *et al.*, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.

[17] P. e. a. Zhu, "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.

[18] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, IEEE, 2011, pp. 3153–3160.

[19] M. Barekatain *et al.*, "Okutama-action: An aerial view video dataset for concurrent human action detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 28–35.

[20] V. Carletti, A. Greco, A. Saggese, and M. Vento, "Multi-object tracking by flying cameras based on a forward-backward interaction," *IEEE Access*, vol. 6, pp. 43 905–43 919, 2018.

[21] H. Fan *et al.*, "Visdrone-mot2020: The vision meets drone multiple object tracking challenge results," in *European Conference on Computer Vision*, Springer, 2020, pp. 713–727.

[22] I. Kalra, M. Singh, S. Nagpal, R. Singh, M. Vatsa, and P. Sujit, "Dronesurf: Benchmark dataset for drone-based face recognition," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–7.

[23] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2118–2125.

[24] P. .-. Chen *et al.*, "Drone-based vehicle flow estimation and its application to traffic conflict hotspot detection at intersections," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1521–1525. DOI: 10.1109/ICIP40778.2020.9190890.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[26] T. Lin *et al.*, "Microsoft COCO: common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[27] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[28] X. Li *et al.*, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *arXiv preprint arXiv:2006.04388*, 2020.

[29] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[30] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 840–849.

[31] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, and L. Shao, "D2det: Towards high quality object detection and instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 485–11 494.

[32] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2778–2788.

[33] W. Yu, T. Yang, and C. Chen, "Towards resolving the challenge of long-tail distribution in uav images for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3258–3267.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[35] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[36] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[37] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[38] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*, Springer, 2014, pp. 391–405.

[39] 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)*, vol. 1, 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[40] W. Liu *et al.*, "SSD: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[41] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[42] H. Wang *et al.*, "Spatial attention for multi-scale feature refinement for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[43] J. Zhang, J. Huang, X. Chen, and D. Zhang, "How to fully exploit the abilities of aerial image detectors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[44] D. R. Pailla, "Visdrone-det2019: The vision meets drone object detection in image challenge results," 2019.

[45] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[46] D. D. et al, "Visdrone-det2020: The vision meets drone object detection in image challenge results," 2020.

[47] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8311–8320.

[48] C. Li, T. Yang, S. Zhu, C. Chen, and S. Guan, "Density map guided object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 190–191.

[49] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2849–2858.

[50] Q. M. Chung, T. D. Le, T. V. Dang, N. D. Vo, T. V. Nguyen, and K. Nguyen, "Data augmentation analysis in vehicle detection from aerial videos," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, IEEE, 2020, pp. 1–3.

[51] Xie, X and Yang W and Cao, G and Yang, J and Shi, G, *The collected xduav dataset.* `https://share.weiyun.com/8rAu3kqr`, Last accessed on 2020-02-13, 2018.

[52] A. Clark, *Pillow (pil fork) documentation*, 2015. [Online]. Available: `https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf`.

[53] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 13 001–13 008.

[54] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[55] S. Hong, S. Kang, and D. Cho, "Patch-level augmentation for object detection in aerial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[56] X. Zhang, E. Izquierdo, and K. Chandramouli, "Dense and small object detection in uav vision based on cascade network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[57] K. Chen *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[58] J. Yang, X. Xie, G. Shi, and W. Yang, "A feature-enhanced anchor-free network for uav vehicle detection," *Remote Sensing*, vol. 12, no. 17, p. 2729, 2020.

Received on April 14, 2023
Accepted on August 28, 2023