TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios

Xingkui Zhu[1] * Shuchang Lyu[1] * Xu Wang [1] Qi Zhao[1] †

[1] Beihang University, Beijing, China

{adlith, lyushuchang, sy2002406, zhaoqi}@buaa.edu.cn

Abstract

Object detection on drone-captured scenarios is a recent popular task. As drones always navigate in different altitudes, the object scale varies violently, which burdens the optimization of networks. Moreover, high-speed and low-altitude flight bring in the motion blur on the densely packed objects, which leads to great challenge of object distinction. To solve the two issues mentioned above, we propose TPH-YOLOv5. Based on YOLOv5, we add one more prediction head to detect different-scale objects. Then we replace the original prediction heads with Transformer Prediction Heads (TPH) to explore the prediction potential with self-attention mechanism. We also integrate convolutional block attention model (CBAM) to find attention region on scenarios with dense objects. To achieve more improvement of our proposed TPH-YOLOv5, we provide bags of useful strategies such as data augmentation, multi-scale testing, multi-model integration and utilizing extra classifier. Extensive experiments on dataset VisDrone2021 show that TPH-YOLOv5 have good performance with impressive interpretability on drone-captured scenarios. On DET-test-challenge dataset, the AP result of TPH-YOLOv5 are 39.18%, which is better than previous SOTA method (DPNetV3) by 1.81%. On VisDrone Challenge 2021, TPH-YOLOv5 wins 5^{th} place and achieves well-matched results with 1^{st} place model (AP 39.43%). Compared to baseline model (YOLOv5), TPH-YOLOv5 improves about 7%, which is encouraging and competitive.

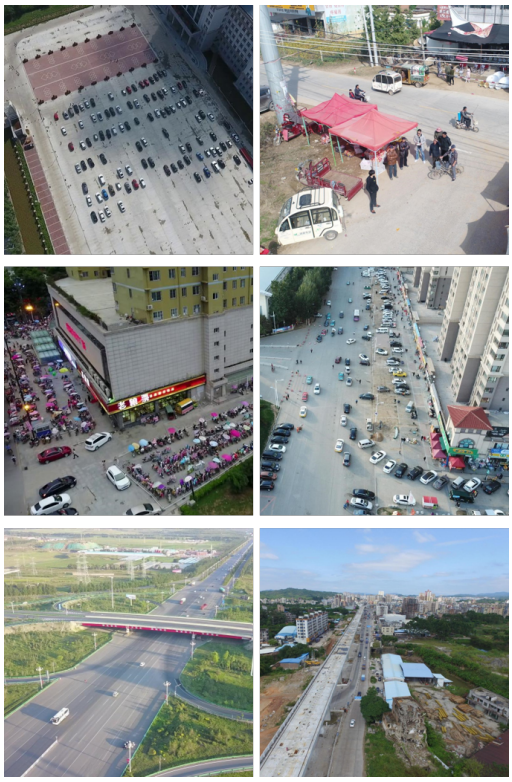

Figure 1. Intuitive cases to explain the three main problems in object detection on drone-captured images. The cases in first row, second row and third row respectively shows the size variation, high-density and large coverage of objects on drone-captured images.

1. Introduction

Object detection technology on drone-captured scenarios has been widely used in many practical applications, such as plant protection [18, 41], wildlife protection [23, 22] and urban surveillance [1, 15]. In this paper, we focus on improving the performance of object detection on drone-captured images and providing insight for the above-mentioned numerous applications.

Recent years have witnessed significant progresses in object detection tasks using deep convolutional neural networks [40, 37, 34, 27, 58]. Some notable benchmark datasets like MS COCO [30] and PASCALVOC [9] greatly

*Contribute Equally.

†Corresponding author.

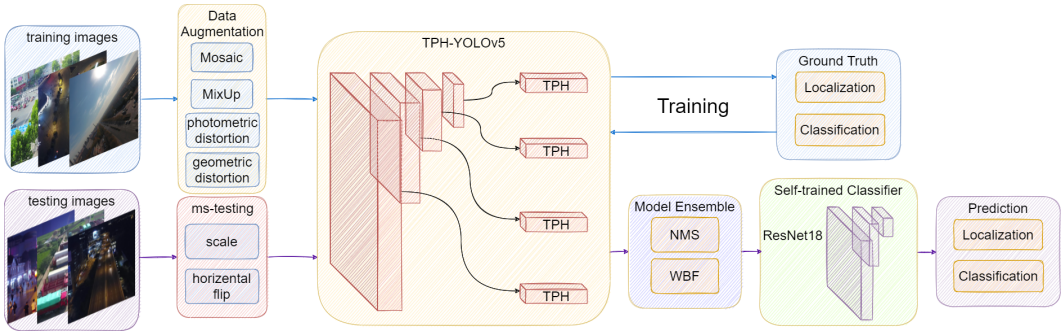

Figure 2. The overview of working pipeline using TPH-YOLOv5. Compared to original version, we mainly improve the head by applying Transformer Prediction Head (TPH). We also add one more head to better detect different scale objects. In addition, we employ bag of tricks like data augmentation, multi-scale testing, model ensemble and self-trained classifier to make TPH-YOLOv5 stronger.

promote the development of object detection application. However, most previous deep convolutional neural networks are designed for natural scene images. Directly applying previous models to tackle object detection task on drone-captured scenarios mainly has three problems, which are intuitively illustrated by some cases in Fig.1. First, the object scale varies violently because the flight altitude of drones change greatly. Second, drone-captured images contain objects with high density, which brings in occlusion between objects. Third, drone-captured images always contain confusing geographic elements because of covering large area. The above-mentioned three problems make the object detection of drone-captured images very challenging.

In object detection task, YOLO series [37, 38, 39, 2] play an important role in one-stage detectors. In this paper, we propose an improved model, TPH-YOLOv5 based on YOLOv5 [21] to solve the above-mentioned three problems. The overview of the detection pipeline using TPH-YOLOv5 is shown in Fig.2. We respectively use CSPDarknet53 [52, 2] and path aggregation network (PANet [33]) as backbone and neck of TPH-YOLOv5, which follows the original version. In the head part, we first introduce one more head for tiny object detection. Totally, TPH-YOLOv5 contains four detection heads separately used for the detection of tiny, small, medium, large objects. Then, we replace the original prediction heads with Transformer Prediction Heads (TPH) [7, 49] to explore the prediction potential. To find the attention region in images with large coverage, we adopt Convolutional Block Attention Module (CBAM [54]) to sequentially generate the attention map along channel-wise and spatial-wise dimensions. Compared to YOLOv5, our improved TPH-YOLOv5 can better deal with drone-captured images.

To further improve the performance of TPH-YOLOv5, we employ bag of tricks (Fig.2). Specifically, we adopt data augmentation during training, which promote the adaptation for dramatic size changes of objects in images. We also add multi-scale testing (ms-testing) and multi-model ensemble strategies during inference to obtain more convincing detection results. Moreover, through visualizing the failure cases, we find that our proposed architecture has excellent localization ability but poor classification ability, especially on some similar categories like "tricycle" and "awning-tricycle". To solve this problem, we provide a self-trained classifier (ResNet18 [17]) using the image patches cropping from training data as classification training set. With self-trained classifier, our method has 0.8%$\sim$1.0% improvement on AP value.

Our contributions are listed as follows:

- We add one more prediction head to deal with large scale variance of objects.
- We integrate the Transformer Prediction Heads (TPH) into YOLOv5, which can accurately localize objects in high-density scenes.
- We integrate CBAM into YOLOv5, which can help the network to find region of interest in images that have large region coverage.
- We provide useful bag of tricks and filtering some useless tricks for object detection task on drone-captured scenarios.
- We use self-trained classifier to improve the classification ability on some confusing categories.
- On VisDrone2021 test-challenge dataset, our proposed TPH-YOLOv5 achieve 39.18% (AP), outperforming DPNetV3 (previous SOTA method) by 1.81%. In VisDrone2021 DET challenge, TPH-YOLOv5 wins 5^{th} place and has minor gap comparing with 1^{st} place models.

2. Related Work

2.1. Data Augmentation

The effectiveness of data augmentation is to expand the dataset, so that the model has higher robustness to the images obtained from different environments. Photometric distortions and geometric distortions are wildly used by researchers. As for photometric distortion, we adjusted the hue, saturation and value of the images. In dealing with geometric distortion, we add random scaling, cropping, translation, shearing, and rotating. In addition to the abovementioned global pixel augmentation methods, there are some more unique data augmentation methods. Some researchers have proposed methods using multiple images together for data augmentation *i.e*. MixUp [57], CutMix [56] and Mosaic [2]. MixUp randomly select two samples from the training images to perform random weighted summation, and the labels of the samples also correspond to the weighted summation. Unlike occlusion works that generally use zero-pixel "black cloth" to occlude a image, CutMix uses an area of another image to cover the occluded area. Mosaic is an improved version of the CutMix. Mosaic stitches four images, which greatly enriches the background of the detected object. In addition, batch normalization calculates the activation statistics of 4 different images on each layer.

In TPH-YOLOv5, we use a combination of MixUp, Mosaic and traditional methods in data augmentation.

2.2. Multi-Model Ensemble Method in Object Detection

Deep learning neural networks are non-linear methods. They provide greater flexibility and can scale in proportion to the amount of training data. One disadvantage of this flexibility is that they learn through random training algorithms, which means that they are sensitive to the details of the training data, and may find a different set of weights each time they train, resulting in different predictions. This gives the neural network a high variance. A successful way to reduce the variance of neural network models is to train multiple models instead of a single model, and combine the predictions of these models.

There are three different methods to ensemble boxes from different object detection models: Non-maximum suppression (NMS) [36], Soft-NMS [53], weighted boxes fusion (WBF) [43]. In the NMS method, if the overlap, intersection over union (IoU) of the boxes is higher than a certain threshold, they are considered to belong to the same object. For each object, NMS only leaves one bounding box with the highest confidence, and other bounding boxes are deleted. Therefore, the box filtering process depends on the choice of this single IoU threshold, which have a big impact on model performance. Soft-NMS has made a slightly change to NMS, which made Soft-NMS shows a significant improvement over traditional NMS on standard benchmark datasets (such as PASCAL VOC [10] and MS COCO [30]). It sets an attenuation function for the confidence of adjacent bounding boxes based on the IoU value instead of completely setting their confidence scores to zero and delete them. WBF works differently from NMS. Both NMS and Soft-NMS exclude some boxes, while WBF merges all boxes to form the final result. Therefore, it can solve all the inaccurate predictions of the model. We use WBF to ensemble final models, which performs much better than NMS.

2.3. Object Detection

CNN-based object detectors can be divided into many types: 1) one-stage detectors: YOLOX [11], FCOS [48], DETR [65], Scaled-YOLOv4 [51], EfficientDet [45]. 2) two-stage detectors: VFNet [59], CenterNet2 [62]. 3) anchor-based detectors: Scaled-YOLOv4 [51], YOLOv5 [21]. 4) anchor-free detectors: CenterNet [63], YOLOX [11], RepPoints [55]. Some detectors are specially designed for Drone-captured images like RRNet [4], PENet [46], CenterNet [63] *etc*. But from the perspective of components, they generally consist of two parts, an CNN-based backbone, used for image feature extraction, and the other part is detection head used to predict the class and bounding box for object. In addition, the object detectors developed in recent years often insert some layers between the backbone and the head, people usually call this part the neck of the detector. Next, we will separately introduce these three structures in detail.

Backbone. The backbone that are often used include VGG [42], ResNet [17], DenseNet [20], MobileNet [19], EfficientNet [44], CSPDarknet53 [52], Swin Transformer [35] *etc*., rather than networks designed by ourselves. Because these networks have proven that they have strong feature extraction capabilities on classification and other issues. But researchers will also fine-tune the backbone to make it more suitable for specific tasks.

Neck. The neck is designed to make better use of the features extracted by the backbone. It reprocesses and rationally uses the feature maps extracted by Backbone at different stages. Usually, a neck consists of several bottom-up paths and several top-down paths. Neck is a key link in the target detection framework. The earliest neck is the use of up and down sampling block. The feature of this method is that there is no feature layer aggregation operation, such as SSD [34], directly follow the head after the multi-level feature map. Commonly used path-aggregation blocks in neck are: FPN [28], PANet [33], NAS-FPN [12], BiFPN [45], ASFF [32], SFAM [61].The commonality of these methods is to repeatedly use various up-and-down sampling, splicing, dot sum or dot product to design aggregation strate-

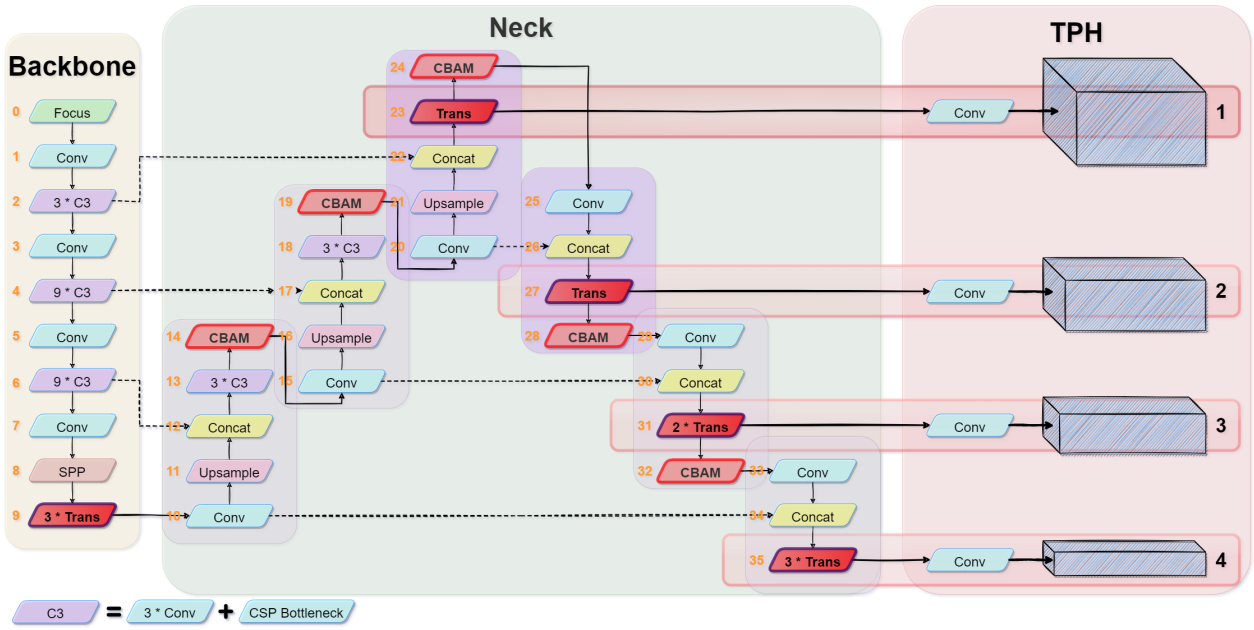

Figure 3. The architecture of the TPH-YOLOv5. a) CSPDarknet53 backbone with three transformer encoder blocks at the end. b) The Neck use the structure like PANet. c) Four TPHs (transformer prediction heads) use the feature maps from transformer encoder blocks in Neck. In addition, the number of each block is marked with orange numbers on the left side of the block.

gies. There are also some additional blocks used in neck, like SPP [16], ASPP [5], RFB [31], CBAM [54].

Head. As a classification network, the backbone cannot complete the positioning task, and the head is designed to be responsible for detecting the location and category of the object by the features maps extracted from the backbone. Heads are generally divided into two kinds: one-stage object detector and two-stage object detector. Two-stage detectors have long been the dominant method in the field of object detection, and the most representative one is the R-CNN series [14, 13, 40]. Compared with the two-stage detector, the one-stage detector predicts the bounding box and the class of objects at the same time. The speed advantage of the one-stage detector is obvious, but the accuracy is lower. For one-stage detectors, the most representative models are YOLO series [37, 38, 39, 2], SSD [34] and RetinaNet [29].

3. TPH-YOLOv5

3.1. Overview of YOLOv5

YOLOv5 has four different models including YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. Generally, YOLOv5 respectively uses the architecture of CSPDarknet53 with an SPP layer as backbone, PANet as Neck and YOLO detection head [37]. To further optimize the whole architecture, bag of freebies and specials [2] are provided. Since it is the most notable and convenient one-stage detector, we select it as our baseline.

When we train the model using VisDrone2021 dataset [64] with data augmentation strategy (Mosaic and MixUp), we find that the results of YOLOv5x are much better than YOLOv5s, YOLOv5m and YOLOv5l, and the gap of AP value is more than 1.5%. Even though the training computation cost of the YOLOv5x model is more than that of other three models, we still choose to use YOLOv5x to pursue the best detection performance. In addition, according to the features of drone-captured images, we adjust the parameters of commonly used photometric distortions and geometric distortions.

3.2. TPH-YOLOv5

The framework of TPH-YOLOv5 is illustrated in Fig. 3. We modify the original YOLOv5 to make it specialize in the VisDrone2021 dataset.

Prediction head for tiny objects. We investigate the VisDrone2021 dataset and find that it contains many extremely small instances, so we add one more prediction head for tiny objects detection. Combined with the other three prediction heads, our four-head structure can ease the negative influence caused by violent object scale variance. As shown in Fig. 3, the prediction head (head No.1) we add is generated from low-level, high-resolution feature map, which is more sensitive to tiny objects. After adding an additional detection head, although the computation and memory cost increase, the performance of tiny objects detection gets large

improvement.

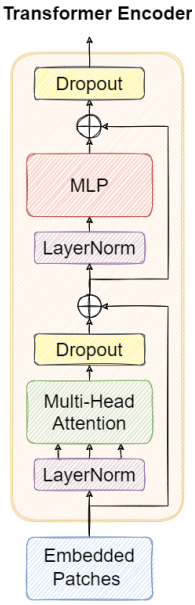

Figure 4. The architecture of transformer encoder, which contains two main blocks, a multi-head attention block and a feed-forward neural network (MLP). LayerNorm and Dropout layers help the network converge better and prevent the network from over fitting. Multi-head attention can help the current node not only pay attention to the current pixels, but also obtain the semantics of the context.

Transformer encoder block. Inspired by the vision transformer [6], we replace some convolutional blocks and CSP bottleneck blocks in original version of YOLOv5 with transformer encoder blocks. The structure is shown in Fig. 4. Compared to original bottleneck block in CSPDarknet53, we believe that transformer encoder block can capture global information and abundant contextual information. Each transformer encoder contains two sub-layers. The first sub-layer is a multi-head attention layer and the second one (MLP) is a fully-connected layer. Residual connections are used between each sub-layer. Transformer encoder blocks increase the ability to capture different local information. It can also explore the feature representation potential with self-attention mechanism [50]. On the VisDrone2021 dataset, transformer encoder blocks have better performance on occluded objects with high-density.

Based on YOLOv5, we only apply transformer encoder blocks in the head part to form Transformer Prediction Head (TPH) and the end of backbone. Because the feature maps at the end of the network have low resolution. Applying TPH on low-resolution feature maps can decrease the expensive computation and memory cost. Moreover, when we enlarge the resolution of input images, we optional remove some TPH blocks at early layers to make the training process available.

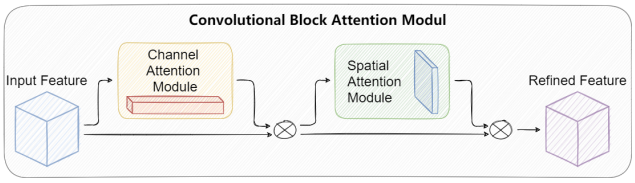

Figure 5. The overview of CBAM module. Two sequential submodules are used to refine feature map that go through CBAM, residual paths are also used.

Convolutional block attention module (CBAM). CBAM [54] is a simple but effective attention module. It is a lightweight module that can be integrated into most notable CNN architectures, and it can be trained in an end-to-end manner. Given a feature map, CBAM sequentially infers the attention map along two separate dimensions of channel and spatial, and then multiplies the attention map with the input feature map to perform adaptive feature refinement. The structure of the CBAM module is shown in the Fig. 5. According to the experiment in the paper [54], after integrating CBAM into different models on different classification and detection datasets, the performance of the model get large improved, which proves the effectiveness of this module.

On drone-captured images, large covering region always contains confusing geographical elements. Using CBAM can extract the attention area to help TPH-YOLOv5 resist the confusing information and focus on useful target objects.

Ms-testing and model ensemble. We train five different models in terms of different perspectives for model ensemble. During inference phase, we first perform ms-testing strategy on single model. The implementation details of ms-testing are the following three steps. 1) Scaling the testing image to 1.3 times. 2) Respectively reducing the image to 1 time, 0.83 times, and 0.67 times. 3) Flipping the images horizontally. Finally, we feed the six different-scaling images to TPH-YOLOv5 and use NMS to fuse the testing predictions.

On different models, we perform the same ms-testing operation and fuse the final five predictions by WBF to get the final result.

Self-trained classifier. After training the VisDrone2021 dataset with TPH-YOLOv5, we test the test-dev dataset and then analyze the results by visualizing the failure cases and draw a conclusion that TPH-YOLOv5 has excellent localization ability but poor classification ability. We further explore the confusion matrix which is shown in Fig.6, and observe that the precision of the some hard categories such as tricycle and awning-tricycle are very low. Therefore, we propose an extra self-trained classifier. First, we construct

a training set by cropping the ground-truth bounding boxes and resizing each image patches to 64×64. Then we select ResNet18 [17] as classifier network. As shown in experimental results, our method get around 0.8%˜1.0% improvement on AP value with the help of this self-trained classifier.

Figure 6. Confusion matrix was made at IoU threshold of 0.45, confidence threshold of 0.25.

4. Experiments

We use the testset-challenge and testset-dev of the VisDrone2021 dataset to evaluate our model, and we report mAP (average of all 10 IoU thresholds, ranging from [0.5: 0.95]) and AP50. VisDrone2021-DET dataset is the same as VisDrone2019-DET dataset and VisDrone2018-DET dataset.

4.1. Implementation Details

We implement TPH-YOLOv5 on Pytorch 1.8.1. All of our models use an NVIDIA RTX3090 GPU for training and testing. In the training phase, we use part of pre-trained model from yolov5x, because TPH-YOLOv5 and YOLOv5 share most part of backbone (block 0˜8) and some part of head (block 10˜13 and block 15˜18), there are many weights can be transferred from YOLOv5x to TPH-YOLOv5, by using these weights we can save a lot of training time.

Because the VisDrone2021 training set is a bit small, we only train the model on VisDrone2021 trainset for 65 epochs, and the first 2 epochs are used for warm-up. We use adam optimizer for training, and use 3e-4 as the initial learning rate with the cosine lr schedule. The learning rate of the last epoch decays to 0.12 of the initial learning rate. The size of the input image of our model is very large, the long side of the image is 1536 pixels, which leads to the batch size is only 2.

Data analysis. According to our previous engineering experience, it is very important to walk through dataset before training the model, which can often be of great help to the improvement of mAP. We have analyzed bounding boxes in the VisDrone2021 dataset. When the input image size is set to 1536, there are 622 of 342391 labels are less than 3 pixels in size. As shown in Fig. 7, these small objects are hard to recognize. When we use gray squares to cover these small objects and train our model on the processed dataset, the mAP improves by 0.2, better than not.

Figure 7. Some images were taken too high, resulting in many small objects, which cannot be recognized.

Ms-testing. When training neural network models for computer vision problems, data augmentation is a technique often used to improve performance and reduce generalization errors. When using a model to make predictions, image data augmentation of test dataset can also be applied to allow the model to make predictions on multiple different versions of images. The prediction of the augmented images can be averaged to get better prediction performance.

We scale the test images to three different sizes in ms-testing, and then flip them horizontally, so that a total of 6 different images are obtained. After testing six different images and fusing the results, we get the final test result.

4.2. Comparisons with the State-of-the-art

On VisDrone2021-DET testset-challenge.

Due to the limited number of submissions in the VisDrone2021 competition server, we only obtained the results of 4 models on testset-challenge and the final results of the ensemble of 5 models. We finally got a good score of 39.18 on testset-challenge, which is much higher than VisDrone2020's best score of 37.37. Ranked fifth in the VisDrone 2021 leader board, our score is 0.25 lower than the 39.43 of the first place. If the number of submissions is not used up, we will definitely get better results. Table 1 lists the score of our model, compared with the scores in the previous year's VisDrone competition and the scores of algorithms submitted by the committee.

Methods	mAP (%)	AP50 (%)
RetinaNet[29]	11.81	21.37
RefineDet[60]	14.90	28.76
DetNet59[26]	15.26	29.23
Cascade-RCNN[3]	16.09	31.91
FPN[28]	16.51	32.20
Light-RCNN[25]	16.53	32.78
CornetNet[24]	17.41	34.12
RRNet (2019 2^{nd})[4]	29.13	55.82
DPNet-ensemble (2019 SOTA) [8]	29.62	54.00
SMPNet (2020 2^{nd})[47]	35.98	59.53
DPNetV3 (2020 SOTA)[47]	37.37	62.05
TPH-YOLOv5 ensemble	**39.18**	\

Table 1. The comparison of the performance in VisDrone2021 testset-challenge

4.3. Ablation Studies

On VisDrone2021-DET testset-dev. we analyze importance of each proposed component on local testset-dev as we cannot test these on VisDrone2021 competition server, the number of submissions to the competition server is very valuable. The impact of each component is listed in the table 2.

Methods	mAP (%)	AP50 (%)
YOLOv5	28.88	49.33
YOLOv5+P2	31.03 (↑2.15)	51.61 (↑ 2.28)
YOLOv5+P2+transformer	32.84 (↑ 1.81)	53.87 (↑ 2.26)
TPH-YOLOv5 (previous+CBAM)	33.63 (↑ 0.79)	54.77 (↑ 0.90)
TPH-YOLOv5+ms-testing	34.90 (↑ 1.27)	56.40 (↑ 1.63)
TPH-YOLOv5+ms-testing+Classifier	35.74 (↑ 0.84)	57.31 (↑ 0.91)

Table 2. Ablation Study on VisDrone2021 testset-dev.

Effect of extra prediction head. Adding a detection head for tiny objects makes the number of layers of the original YOLOv5x change from 607 to 719, and GFLOPs from 219.0 to 259.0. This of course increases the amount of calculation, but the mAP improvement is also very high. From Fig. 9 we can see that TPH-YOLOv5 performs well when detecting small objects, so the increasing in calculation is worthwhile.

Effect of transformer encoder blocks. After using the transformer encoder block, the total layers of the model decrease from 719 to 705, and GFLOPs from 259.0 to 237.3. Use transformer encoder blocks can not only increase mAP, but also reduce the size of the network. At the same time, it also plays a role in the detection of dense objects and large objects.

Effect of model ensemble. We list the mAP of the final results of our five different models in each category and compared them with the fusion model in table 3. In training phrase, we use different input image sizes and change the weight of each category to make each model unique. So that the final ensemble model can get a relatively balanced result. 1) TPH-YOLOv5-1 use the input image size of 1920 and all categories have equal weights. 2) TPH-YOLOv5-2 use the input image size of 1536 and all categories have equal weights. 3) TPH-YOLOv5-3 use the input image size of 1920 and the weight of each category is related to the number of labels, which is shown in Fig. 8. The more labels of a certain category, the lower the weight it is given. 4) TPH-YOLOv5-4 use the input image size of 1536 and the weight of each category is related to the number of labels. 5) TPH-YOLOv5-5 use the backbone of YOLOv5l and use the input image size of 1536.

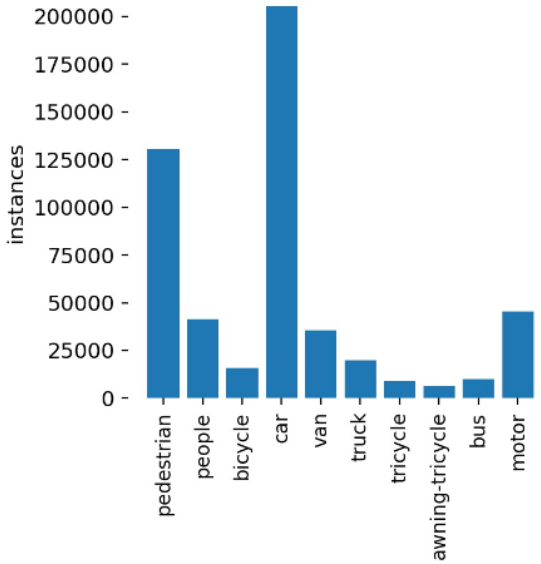

Figure 8. The number of labels of each category.

Some detection result on VisDrone2021 testset-challenge. We have selected some representative images as the display of the test results. Fig. 9 shows the result of large objects, tiny objects, dense objects and the image covering a large area.

5. Conclusion

In this paper, we add some cutting-edge techniques *i.e*. transformer encoder block, CBAM and some experienced tricks to YOLOv5 and form a state-of-the-art detector called TPH-YOLOv5, which is especially good at object detection in drone-captured scenarios. We refresh the record of VisDrone2021 dataset, our experiments showed that TPH-YOLOv5 achieved state-of-the-art performance in VisDrone2021 dataset. We have tried a large number of features, and used some of them to improve the accuracy of object detector. We hope this report can help developers and researchers get a better experience in the analysis and processing of drone-captured scenarios.

Methods	all	pedestrian	people	bicycle	car	van	trunk	tricycle	awning-tricycle	bus	motor
TPH-YOLOv5-1	34.90	27.52	15.32	15.21	65.99	44.23	47.56	23.96	22.11	58.85	28.44
TPH-YOLOv5-2	34.29	27.97	14.88	14.17	67.63	45.01	44.76	25.12	20.48	55.72	27.74
TPH-YOLOv5-3	34.68	22.88	16.01	19.26	48.88	42.98	47.82	32.86	35.65	54.16	28.25
TPH-YOLOv5-4	34.17	23.48	15.79	17.62	49.99	42.76	47.13	31.66	32.21	54.19	27.37
TPH-YOLOv5-5	33.04	25.98	14.90	13.10	63.05	43.45	42.56	25.20	21.06	53.65	27.10
TPH-YOLOv5 ensemble	37.32	29.00	16.75	15.69	68.94	49.79	45.16	27.33	24.72	61.80	30.90

Table 3. Comparison of TPH-YOLOv5 models‘ performances on VisDrone2021 testset-dev for each category.

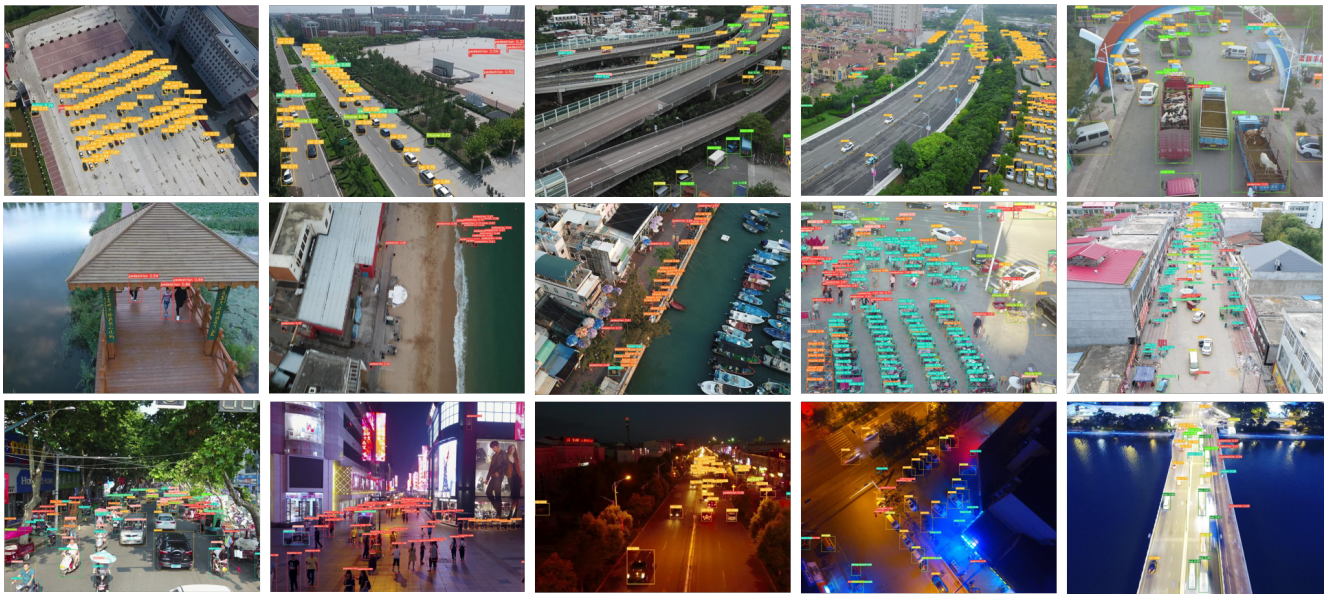

Figure 9. Some visualization results from our TPH-YOLOv5 on testset-challenge, different category use bounding boxes with different color. The performance is good at localization tiny objects, dense objects and objects blurred by motion.

6. Acknowledgments

This work was supported by National Natural Science Foundation of China (62072021).

References

[1] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:20–32, 2018.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[4] Changrui Chen, Yu Zhang, Qingxuan Lv, Shuo Wei, Xiaorui Wang, Xin Sun, and Junyu Dong. Rrnet: A hybrid detector for object detection in drone-captured images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[8] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Lin, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*

Workshops, pages 0–0, 2019.

[9] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2019.

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] Jingjing Gu, Tao Su, Qiuhong Wang, Xiaojiang Du, and Mohsen Guizani. Multiple moving targets surveillance based on a cooperative network for multi-uav. *IEEE Commun. Mag.*, 56(4):82–89, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Jennifer N. Hird, Alessandro Montaghi, Gregory J. McDermid, Jahan Kariyeva, Brian J. Moorman, Scott E. Nielsen, and Anne C. S. McIntosh. Use of unmanned aerial vehicles for monitoring recovery of forest vegetation on petroleum well sites. *Remote. Sens.*, 9(5):413, 2017.

[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[21] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, Apr. 2021.

[22] Benjamin Kellenberger, Diego Marcos, and Devis Tuia. Detecting mammals in uav images: Best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139–153, 2018.

[23] Benjamin Kellenberger, Michele Volpi, and Devis Tuia. Fast animal detection in UAV images using convolutional neural networks. In *2017 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2017, Fort Worth, TX, USA, July 23-28, 2017*, pages 866–869. IEEE, 2017.

[24] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[25] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017.

[26] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Detnet: A backbone network for object detection. *arXiv preprint arXiv:1804.06215*, 2018.

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017.

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[31] Songtao Liu, Di Huang, et al. Receptive field block net for accurate and fast object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 385–400, 2018.

[32] Songtao Liu, Di Huang, and Yunhong Wang. Learning spatial fusion for single-shot object detection. *arXiv preprint arXiv:1911.09516*, 2019.

[33] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[36] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[38] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[39] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[41] Zhenfeng Shao, Congmin Li, Deren Li, Orhan Altan, Lei Zhang, and Lin Ding. An accurate matching method for projecting vector data into surveillance video to monitor and protect cultivated land. *ISPRS Int. J. Geo Inf.*, 9(7):448, 2020.

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[43] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021.

[44] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[45] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[46] Ziyang Tang, Xiang Liu, Guangyu Shen, and Baijian Yang. Penet: object detection using points estimation in aerial images. *arXiv preprint arXiv:2001.08247*, 2020.

[47] Visdrone Team. Visdrone 2020 leaderboard. Website, 2020. `http://aiskyeye.com/visdrone-2020-leaderboard/`.

[48] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[51] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13029–13038, 2021.

[52] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.

[53] Derui Wang, Chaoran Li, Sheng Wen, Qing-Long Han, Surya Nepal, Xiangyu Zhang, and Yang Xiang. Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples. *IEEE Transactions on Cybernetics*, 2021.

[54] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[55] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9657–9666, 2019.

[56] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[58] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021.

[59] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021.

[60] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018.

[61] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9259–9266, 2019.

[62] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021.

[63] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[64] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.