2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

HIC-YOLOv5: Improved YOLOv5 For Small Object Detection

Shiyi Tang, Shu Zhang, Yini Fang
Heriot-Watt University, Ocean University of China, Hong Kong University of Science and Technology
st2015@hw.ac.uk, zhangshu@ouc.edu.cn, yfangba@connect.ust.hk

***Abstract*— Small object detection has been a challenging problem in the field of object detection. There has been some works that proposes improvements for this task, such as adding several attention blocks or changing the whole structure of feature fusion networks. However, the computation cost of these models is large, which makes deploying a real-time object detection system unfeasible, while leaving room for improvement. To this end, an improved YOLOv5 model: HIC-YOLOv5 is proposed to address the aforementioned problems. Firstly, an additional prediction head specific to small objects is added to provide a higher-resolution feature map for better prediction. Secondly, an involution block is adopted between the backbone and neck to increase channel information of the feature map. Moreover, an attention mechanism named CBAM is applied at the end of the backbone, thus not only decreasing the computation cost compared with previous works but also emphasizing the important information in both channel and spatial domain. Our result shows that HIC-YOLOv5 has improved mAP@[.5:.95] by 6.42% and mAP@0.5 by 9.38% on VisDrone-2019-DET dataset.**

I. INTRODUCTION

Object detection algorithm has been widely applied to smart systems of Unmanned Aerial Vehicles (UAVs), such as pedestrian detection and vehicle detection. It automates the analysis process of the photos taken by UAVs. However, the biggest issue of such applications lies in detecting small objects, as most of the objects in the photos become smaller from a higher altitude. This fact poses negative effects on the accuracy of object detection, including target occlusion, low target density, and dramatic changes in light.

You Only Look Once (YOLO) [15] is a commonly used one-stage object detection algorithm. It takes an image as input and outputs the information of the objects in one stage. Among YOLO series, YOLOv5 [21] is dominating real-time UAV systems due to its low latency and high accuracy compared to other YOLO versions. However, it still has drawbacks in UAV scenarios with a large number of small objects. To address this issue, there have been previous works to improve the performance of small object detection. Some works [7][9] improve the whole structure of the feature fusion network of YOLOv5. Other works [11] add several attention blocks in the backbone. However, the computation cost is large among the previous methods and there is still improvement space of the performance.

In this paper, we propose an improved YOLOv5 algorithm: **HIC-YOLOv5** (**H**ead, **I**nvolution and **C**BAM-**YOLOv5**) for small object detection, with better performance and less computation cost. We first add an additional prediction head, Small Object Detection Head (SODH), dedicated to detecting small objects from feature maps with a higher resolution. The features of tiny and small objects are more easily extracted when the resolution of the feature map increases. Secondly, we add a Channel feature fusion with involution (CFFI) between the backbone and the neck to enhance the channel information, thus enabling more information transmitted to the deep network. Finally, we apply a lightweight Convolutional Block Attention Module (CBAM) at the end of the backbone, which not only has a lower computation cost than [11] but also improves the total performance by emphasizing important channel and spatial features. The experiment result shows that our HIC-YOLOv5 has improved the performance of YOLOv5 on VisDrone dataset by 6.42% (mAP@[.5:.95]) and 9.38% (mAP@0.5) compared to the baseline.

Our main contributions can be summarized as follows:

- The additional prediction head is designed especially for small objects. It detects objects in higher-resolution feature maps, which contain more information about tiny and small objects.
- An involution block is added as a bridge between the backbone and neck to increase the channel information of the feature maps.
- CBAM is applied at the end of the backbone, thus more essential channel and spatial information is extracted while the redundant ones are ignored.

II. RELATED WORK

A. Object Detection Primer

The main purpose of object detection is to locate and classify objects in images, in the form of bounding boxes and confidence scores labeled on the objects. There are two types of object detection pipelines: two-stage and one-stage detectors. The two-stage detectors (e.g., R-CNN[1], SPP-net[2], Fast R-CNN[3] and FPN[4]) first generate region proposals, and then apply object classification and height and width regression. The one-stage detectors (e.g., YOLO series and SSD[16]) use a deep learning model, which directly takes an image as input and outputs bounding box coordinates and class probabilities.

Among all the YOLO series, YOLOv5 [21] is the most suitable algorithm for real-time object detection due to its promising performance and excellent computational efficiency. There have been several versions of YOLOv5, with

*This work was not supported by any organization

[1]Shiyi Tang Author majors in Computer Science, Ocean University of China & Heriot-Watt University st2015@hw.ac.uk

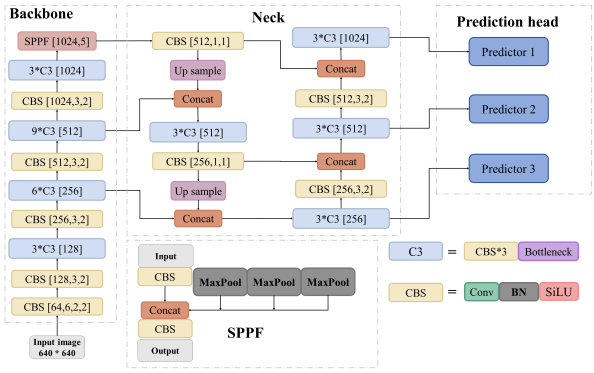


Fig. 1. Structure of YOLOv5-6.0.

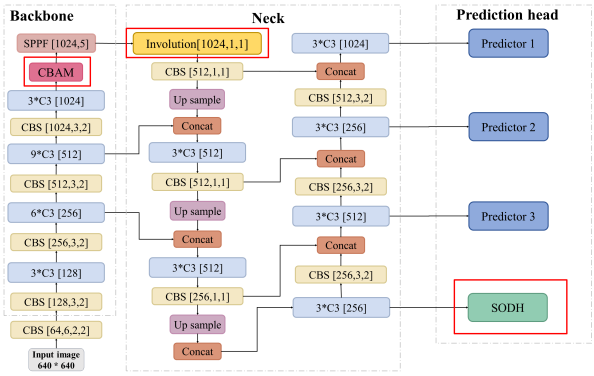


Fig. 2. Structure of HIC-YOLOv5.

the same main structure but a few differences on some small modules. Our model is based on YOLOv5-6.0, and can be easily extended to other versions by applying the proposed modules. The main structure of YOLOv5-6.0 are shown in Fig. 1. The detailed algorithm can be found in [21].

There are totally 3 prediction heads with input size 80 $\times$ 80, 40 $\times$ 40, 20 $\times$ 20 in YOLOv5 for detecting three types of object sizes respectively: small, medium, and large. The coarser (smaller) input feature map represents features for larger objects. Different from images in other natural scenes, objects in UAV systems can be as tiny as only 0.1% of the images. Consequently, we designed a new head with a larger input feature size for smaller objects.

B. Previous Works on Small Object Detection

There have been numerous prior efforts aimed at enhancing the detection performance of small objects. Certain studies have focused on optimizing the neck component in YOLOv5. For instance, [9] replaces the PANet with a weighted bidirectional feature pyramid Mul-BiFPN and [7] introduces a new feature fusion method PB-FPN to the neck. However, both methods choose to change the entire structure of neck to achieve better feature fusion, which results in larger computation cost. Instead, we introduce a lightweight involution block between the backbone and the neck without changing the neck architecture, achieving less computation cost and higher accuracy. [6] introduces a spatio-temporal interaction module, which applies recursive gated convolution to make greater spatial interaction, but causes channel information loss because of 1 $\times$ 1 convolution layers. In our structure, the involution block can effectively address this problem by sharing information across channels using a specific involution kernel and incorporating information of both spatial and channel.

Attention mechanisms have been widely applied in the field of computer vision, which learns to emphasize essential parts and ignore the unimportant ones in an image. There are various types of attention mechanisms, such as channel attention, spatial attention, temporal attention, and branch attention. [14] uses CBAM to obtain both spatial and channel attention. [19] applies a vision transformer to capture the relationship between global and local features in an image. [11] adds a transformer layer at the end of the backbone. However, it requires large computation costs and is difficult to train the model when the input image size is large. Compared with these methods, we add a lightweight CBAM block at the end of the backbone, which aims to use less computation cost and focus more on essential information when extracting features.

III. METHODOLOGY

The structure of HIC-YOLOv5 is shown in Fig. 2. The initial YOLOv5 consists of 3 sections: backbone for feature extraction, neck for feature fusion, and 3 prediction heads. Based on the default model, we propose three modifications: 1) we add an additional prediction head to detect layers with high-resolution feature maps for small and tiny objects specifically; 2) an Involution block is adopted at the beginning of the neck to improve the performance of PANet; 3) we incorporate the Convolutional Block Attention Module (CBAM) into the backbone network.

A. Convolutional Block Attention Module(CBAM)

Previous works add CBAM into the neck block [10]. However, the parameters and computing cost increase because some feature maps connected with CBAM have large sizes. Moreover, the model is difficult to train due to the large amount of parameters. Hence, we adopt CBAM in the Backbone network to highlight significant features when extracting features in the backbone, rather than generating feature pyramids in the neck. The feature map size as the input of CBAM is only 20 $\times$ 20 which is 32 times smaller than the 640 $\times$ 640 full image so the computing cost will not be large.

CBAM is an effective model based on the attention mechanism, which can be conveniently integrated into CNN architectures. It consists of 2 blocks: Channel Attention Module and Spatial Attention Module, as shown in Fig. 3. The two modules respectively generate a channel and a spatial attention map, which are then multiplied with the input feature map to facilitate adaptive feature refinement. Therefore, the meaningful features along both channel and spatial axes are emphasized, while the redundant ones are suppressed. The Channel Attention Module performs global

Max-pooling and Average-pooling for feature maps on different channels and then executes element-wise summation and sigmoid activation. The Spatial Attention Module performs a global Max-pooling and Average-pooling for Values of pixels in the same position on different feature maps and then concatenates the two feature maps, followed by a Conv2d operation and sigmoid activation.

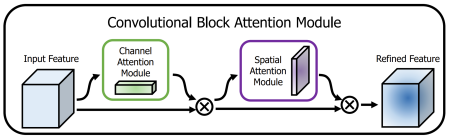


Fig. 3. Structure of CBAM[14].

B. Channel Feature Fusion with Involution (CFFI)

The neck of YOLOv5 adopts PANet, which introduces the bottom-up path augmentation structure on the basis of FPN. The corresponding structure of FPN and bottom-up path augmentation is shown in Fig. 4. Particularly, FPN has great ability to detect small and tiny targets by fusing features of high and low layers so as to obtain high resolution and strong semantics features. However, a 1×1 convolution is adopted to reduce the number of channels at the beginning of the neck in the initial YOLOv5 architecture, where the calculation efficiency is significantly improved, but the channel information is also reduced, leading to poor performance of PANet. Inspired by [11], we add an Involution block between the backbone and the neck. The channel information is improved and shared, resulting in the reduction of information loss during the initial phases of FPN. As a result, this improvement contributes to the enhanced performance of FPN, particularly benefiting the detection of objects with smaller sizes. Moreover, it is emphasized that Involution has better adaptation to various visual patterns in terms of different spatial positions.[13]

The structure of Involution is illustrated in Fig. 5. Involution kernels, represented as $\mathscr{H} \in \mathbb{R}^{H\times W\times K\times K\times G}$, are designed to incorporate transformations that exhibit inverse attributes in both the spatial and channel domains, where H and W represents the height and width of the feature map, K is the kernel size and G represents the number of groups, where each group shares the same involution kernel. Particularly, a specific involution kernel, denoted as $\mathscr{H}_{i,j,\cdot,\cdot,g} \in \mathbb{R}^{K\times K}, g = 1,2,...,G$, is designed for the pixel

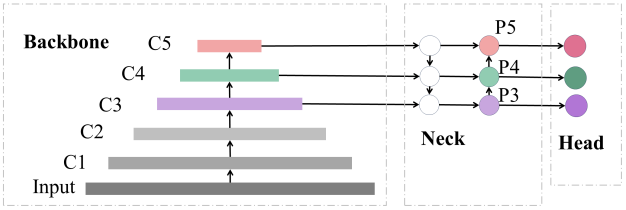


Fig. 4. FPN and PANet structure in YOLOv5.

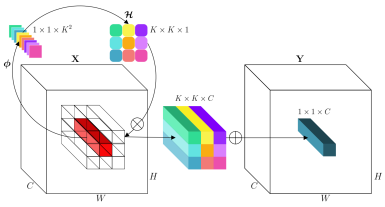


Fig. 5. Structure of Involution block[13].

$\mathbf{X}_{i,j} \in \mathbb{R}^C$ (the subscript of C is omitted for brevity), while being shared across the channels. Finally, the output feature map of involution $\mathbf{Y}_{i,j,k}$, is obtained as follows:

$$\mathbf{Y}_{i,j,k} = \sum_{(u,v)\in\Delta_K} \mathscr{H}_{i,j,u+\lfloor K/2\rfloor,v+\lfloor K/2\rfloor,\lceil kG/C\rceil}\mathbf{X}_{i+u,j+v,k} \quad (1)$$

Therefore, the information contained in the channel dimension of a single pixel is implicitly dispersed to its spatial vicinity, which is useful to obtain the enriched receptive field information.

C. Prediction Head

The different resolutions (80×80, 40×40, and 20×20) of 3 prediction heads in YOLOv5 make a great contribution to the detection ability in various application scenarios, but also make it difficult to detect small and tiny objects. The areas of tiny objects only contain a few pixels, which makes the model difficult to learn, and convolutional blocks further reduce the resolution of feature maps when the depth of the network increases. In order to solve this issue and inspired by [9][12][6], we propose an additional prediction head, Small Object Detection Head (SODH), which aims to detect feature maps with larger resolution (160×160). The larger resolution the feature map, the more detailed information is obtained, making it increasingly effortless to extract features from small and tiny objects.

Each prediction head takes the feature extracted and fused by backbone and neck as input, and finally outputs a vector, which consists of the regression bounding box (coordinate and size), the confidence of the object's border and the class of the object. Before generating the final bounding boxes, we generate anchors to form the candidate bounding boxes. These anchors are generated by k-means according to the dataset and are defined in 3 different scales for the 3 prediction heads, adapting to small, middle and large objects respectively. Anchors of the additional prediction head are also generated by k-means.

D. Loss Function

The loss function of HIC-YOLOv5 consists of three sections: objectness, bounding box and class probability, which can be represented as follows:

$$\text{Loss} = \alpha\text{Loss}_{obj} + \beta\ \text{Loss}_{\ box} + \gamma\text{Loss}_{cls} \quad (2)$$

We use binary cross entropy loss for both objectness and class probability, and CIoU loss [20] for bounding box regression.

E. Data Augmentation

We applied the data augmentation used in the initial YOLOv5 (e.g., Mosaic, Copy paste, Random affine, MixUp, HSV augmentation and Cutout) Additionally, we found that many small people and cars are in the center of a picture in the dataset. Therefore we add extra center cropping.

IV. EXPERIMENTAL RESULTS

A. Experimental Setting

1) Experimental Equipment: In this experiment, the CPU is 15 vCPU Intel® X Platinum 8358P CPU @ 2.60GHz, the GPU is NVIDIA A40 with 48 GB of Graphics memory. The algorithm is implemented by PyTorch, using CUDA 11.6 to operation acceleration.

2) Dataset: We use VisDrone2019 [22] for the evaluation. It was collected by the AISKYEYE team at Lab of Machine Learning and Data Mining, Tianjin University, China. It comprises 288 video clips consisting of 261,908 frames and 10,209 static images captured by diverse drone-mounted cameras across different locations separated by thousands of kilometers in China, environments containing both urban and rural, objects including pedestrians, vehicles, bicycles etc., and densities from sparse to crowded scenes. Notably, this dataset was acquired using multiple drone platforms with varying models under different scenarios as well as weather and lighting conditions. The dataset is divided into a training set, a validation set and a testing set, with 6471, 548, 1610 images respectively.

The detailed information of the dataset is visualized in Fig 6. There are totally 10 classes in this dataset (e.g., pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor) as shown in 6a. It can be observed from the 6c and TABLE I that 75% objects are 0.001 times smaller than the image size, indicating the large number of small and tiny objects. There are 2.5% objects larger than 0.01 times of the image size. Therefore we keep the heads for medium and large objects.

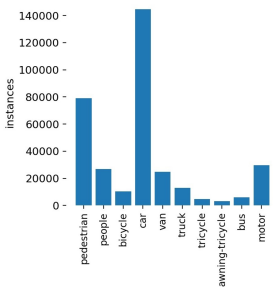
TABLE I
AREA OF OBJECTS RELATIVE TO THE ORIGINAL IMAGE SIZE

mean	std	min	25%	50%	75%	max
0.001535	0.003852	0	0.00017	0.000462	0.001342	0.302962

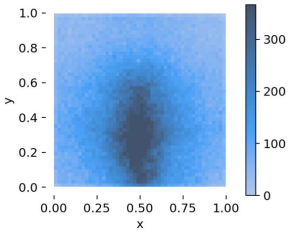
B. Hyperparemter Settings

In order to accelerate the training speed, the input image size is set to be 640 × 640. We set the batch size to be 128 and training epoch to be 300. We use early stopping strategy to avoid over-fitting, where the patience is set to 15. We use Adam as the optimizer, with an initial learning rate of 0.001. Other detailed parameters are listed in Table II. The weights of the loss function are set to be 0.5 (object), 0.05 (box) and 0.25 (class) respectively.

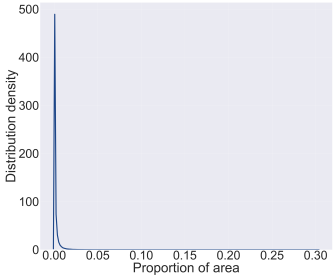
There are mainly 5 models in YOLOv5, including YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. The depth and width of the models increase in sequence while other structures stay the same. The larger the model,



(a) Instances. Information of 10 classes in VisDrone-2019.



(b) Locations. The locations of objects in an image, and heights and widths are normalized to be between 0 and 1.



(c) Distribution. The distribution of object areas in VisDrone-2019.

Fig. 6. Dataset.

the more precise the result. However, in order to accelerate the training speed, we choose to use YOLOv5s during the experiment, with a depth and width of 0.33 and 0.50 respectively.

Data augmentation parameters are listed in Table II. In the center cropping, the height and width of center crop are set to be half of the original image size.

TABLE II
PARAMETER SETTINGS.

hsv_h	hsv_s	hsv_v	degrees	scale	mosaic	mixup	copy_paste
0.4	0.3	0.5	0.2	0.4	1	0.2	0.1

Some default anchors are predefined for coco data sets. Before the training starts, annotation details in the dataset will be examined automatically and the most suitable recall rate for the default anchor will be computed. If the optimal recall rate equals or exceeds 0.98, it is not necessary to update the anchor frame. However, YOLOv5 will recalculate the anchors if the optimal recall rate falls below 0.98. During this experiment, 4 groups of anchors of 4 prediction heads are listed in TableIII. Each group is applied for different sizes of feature maps. Specifically, there are 3 pairs of anchors in each group for a single ground truth. Therefore, the overall number of anchors is 4 × 3=12.

C. Evaluation criterion

mAP is used to evaluate the performance of our improved algorithm. The detailed definitions are listed below.

TABLE III

ANCHOR SIZES FOR PREDICTION HEADS OF HIC-YOLOV5.

Detection head	Anchor frame size
Tiny	[2.9434,4.0435], [3.8626,8.5592], [6.8534, 5.9391]
Small	[10,13], [16,30], [33,23]
Medium	[30,61], [62,45], [59,119]
Large	[116,90], [156,198], [373,326]

(a) YOLOv5 (b) HIC-YOLOv5

Fig. 7. Comparison of detection effect between YOLOv5s and HIC-YOLOv5.

1) mAP: The Average Precision (AP) is a measure of the Precision scores at different thresholds along the Precision-Recall (PR) curve, and is calculated as a weighted mean. Mean Average Precision (mAP) is the mean values of the AP for all classes. Specifically, mAP@0.5 represents the mAP when IoU is 0.5, mAP@[.5:.95] is the mean mAP when IoU ranges from 0.5 to 0.95.

D. Experimental Results

TABLE IV

COMPARISON OF ALGORITHMS ON VISDRONE2019 DATASET.

Method	Dataset	mAP@.5	mAP@[.5:.95]
YOLOv5	Test	27.57	14.43
Shang et al. [5]	Test	36.4	20.1
Liu et al. [6]	Test	35.3	20
Liu et al. [7]	Test	34.3	18.2
Ding et al. [8]	Val	42.9	24.6
HIC-YOLOv5	Test	36.95	20.85
HIC-YOLOv5	Val	44.31	25.95

From Table IV, we can see that compared with the YOLOv5s model, the mAP@[.5:.95] has been improved by 6.42% and mAP@0.5 has been improved by 9.38%. The small object detection head greatly helps to retain the features of small objects. Additionally, Involution effectively amplifies the channel information, while the CBAM block selectively emphasizes crucial features during their extraction within the backbone. The detection effect between YOLOv5s and HIC-YOLOv5 is shown in Fig 7. It visually indicates that more small objects can be detected when using the improved method.

We also compared our improved model with other algorithms tested on VisDrone2019. The results are shown in Table IV. It can be seen that our proposed model has greater performance compared to other detection models. For instance, [6] applies recursive gated convolution to make greater spatial interaction, but it can lead to the loss of channel information due to 1×1 convolution layers. Compared with [6], we adopt involution block to enhance the channel information of PANet, thus improving the performance of small object detection. [11] adds a transformer layer at the end of the backbone, which has the shortcoming of large computation cost. Instead, we apply a lightweight CBAM block, which decreases the training time and computational cost. The number of layers, parameters, gradients and latency of [11] and YOLOv5+CBAM are listed in TABLE V. The mAP is incomparable since [11] uses another different dataset. We use image of 640×640 and calculate the average latency of 100 runs. It can be observed that the CBAM is faster than a transformer layer.

TABLE V

COMPUTATIONAL COMPARISON OF WANG ET AL.[11] AND YOLOV5+CBAM.

Model	Layers	Parameters	Latency(ms)
Wang et al. [11]	297	14580167	456.4727
YOLOv5+cbam	289	8391641 (57.56% of [11])	435.5298

E. Ablation Study

We conducted several experiments to study the effect of three modifications: additional prediction head, involution block and CBAM. The results of ablation study are shown in TableVI. It can be observed that the fourth prediction head makes best contribution to the performance of the model, which improved mAP@.5 and mAP@[.5:.95] by 8.31% and 5.51% respectively. Instead, the single block of CBAM and Involution could not improve the model without the help of fourth prediction head. We speculate that it is because there are so many small objects in VisDrone2019 dataset that the single block of CBAM and Involution can not perform well if these small objects can not be detected first. Based on adding the fourth prediction head, the involution block also has great improvement on the model, with 0.66% and 0.57% increase of mAP@.5 and mAP@[.5:.95]. Additionally, mAP@.5 and mAP@[.5:.95] are improved by 0.41% and 0.34% respectively.

We also compared the latency of different YOLO versions with our proposed modules in VII with same input image size 640×640. YOLOv5 is faster than YOLOv7 and YOLOv8, but slower than YOLOv6. Our modules can be plugged into any YOLO version easily, and one can choose the version based on the task formulation or the requirement of computing resources.

V. CONCLUSIONS

In this paper, an improved YOLOv5 algorithm HIC-YOLOv5 has been proposed, aiming to improve the performance of small and tiny object detection. There are three main contributions in this paper and the experimental results

TABLE VI

ABLATION STUDY

Model	P	R	mAP@.5	mAP@[.5:.95]
Baseline	37.59	31.52	27.57	14.43
+SODH	46.18	37.89	35.88	19.94
+cbam	36.13	28.16	24.2	11.92
+involution	35.49	31.13	26.76	13.8
+SODH+cbam	44.5	35.35	34.53	19.03
+SODH+involution	46.48	37.91	36.54	20.51
+SODH+involution+cbam	47.88	38.49	36.95	20.85

TABLE VII

LATENCY OF DIFFERENT YOLO VERSIONS WITH PROPOSED MODULES

YOLO version	Latency (ms)
YOLOv5 [21]	585.73
YOLOv6 [23]	471.61
YOLOv7 [24]	1455.07
YOLOv8 [25]	710.70

has proved the effectiveness of our methodology. Firstly, an additional prediction head for small objects is added so that the higher-resolution feature maps could be directly used to detect small targets. Secondly, we adopt an involution block between the backbone and neck, thus increasing channel information of the feature map. Furthermore, we also apply an attention mechanism named CBAM at the end of the backbone to decrease the computation cost and emphasize the important information in both channel and spatial domain. Additionally, data augmentation such as center crop is also applied apart from the original data augmentation methods in YOLOv5. Therefore, the improved YOLOv5 is able to increase the accuracy of detecting small and tiny objects.

REFERENCES

[1] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014. [Online]. Available: http://arxiv.org/abs/1406.4729

[3] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

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

[5] J. Shang, J. Wang, S. Liu, C. Wang, and B. Zheng, "Small target detection algorithm for uav aerial photography based on improved yolov5s," *Electronics*, 2023.

[6] H. Liu, X. Duan, H. Lou, J. Gu, H. Chen, and L. Bi, "Improved gbs-yolov5 algorithm based on yolov5 applied to uav intelligent traffic," *Scientific Reports*, vol. 13, no. 1, p. 9577, Jun 2023. [Online]. Available: https://doi.org/10.1038/s41598-023-36781-2

[7] H. Liu, F. Sun, J. J. Gu, and L. Deng, "Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors*, vol. 22, no. 5817, 2022.

[8] K. Ding, X. Li, W. Guo, and L. Wu, "Improved object detection algorithm for drone-captured dataset based on yolov5," in *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2022, pp. 895–899.

[9] J. Shang, J. Wang, S. Liu, C. Wang, and B. Zheng, "Small target detection algorithm for uav aerial photography based on improved yolov5s," *Electronics*, 2023.

[10] J. An, M. D. Putro, A. Priadana, and K.-H. Jo, "Improved yolov5 network with cbam for object detection vision drone," in *2023 IEEE International Conference on Industrial Technology (ICIT)*, 2023, pp. 1–6.

[11] A. Wang, T. Peng, H. Cao, Y. Xu, X. Wei, and B. Cui, "Tia-yolov5: An improved yolov5 network for real-time detection of crop and weed in the field," *Frontiers in Plant Science*, vol. 13, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2022.1091655

[12] 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," 2021.

[13] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, and Q. Chen, "Involution: Inverting the inherence of convolution for visual recognition," 2021.

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," 2018.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. [Online].

[17] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 346–361. [Online].

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[20] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," 2021.

[21] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, NanoCode012, TaoXie, Y. Kwon, K. Michael, L. Changyu, J. Fang, A. V, Laughing, tkianai, yxNONG, P. Skalski, A. Hogan, J. Nadar, imyhxy, L. Mammana, AlexWang1900, C. Fati, D. Montes, J. Hajek, L. Diaconu, M. T. Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106, "ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support," Oct. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5563715

[22] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

[23] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "Yolov6: A single-stage object detection framework for industrial applications," 2022.

[24] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.

[25] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics