# UAV Lightweight Object Detection Based on the Improved YOLO Algorithm

YunFei Chen†
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology & Fujian Society of Aeronautics and Astronautics
Fuzhou, China
316233632@qq.com

Xingwu Chen
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology & Fujian Society of Aeronautics and Astronautics
Fuzhou, China
cxw@fjut.edu.cn

Lingfeng Chen
CTFF Information Technology Co., Ltd
Fuzhou, China
chenlingfeng@ffcs.cn

Dongwei He
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology & Fujian Society of Aeronautics and Astronautics
Fuzhou, China
He_dw@fjut.edu.cn

Jishi Zheng
School of Transportation
Fujian University of Technology & Fujian Society of Aeronautics and Astronautics
Fuzhou, China
zhengjishi@fjut.edu.cn

Chao Xu
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology & Fujian Society of Aeronautics and Astronautics
Fuzhou, China
1782725439@qq.com

Yang Lin
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology
Fuzhou, China
923761893@qq.com

Lisang Liu
School of Electronic, Electrical Engineering and Physics
Fujian University of Technology
Fuzhou, China
liulisang@fjut.edu.cn

## Abstract

Aiming at the characteristics of small objects in low-altitude images, special shooting angles, and variable shooting angles of unmanned aerial vehicles (UAVs), this paper proposes a structural innovation based on the YOLOv5-MobileNetv3Small network model. It transplants the MobileNetv3 network structure and improves the BackBone network structure to solve the problem of inference high-pixel images taking up too much memory for low-power edge computing nodes. The improved YOLO algorithm uses the Visdrone2019-DET dataset to train the network model and uses the Jetson NX edge computing platform on the UAV to process 1920*1080 video streams for testing. The memory usage of the optimized YOLOv5-MobileNetv3Small network model can be reduced by 72.4%.


*EITCE 2021,* October 22–24, 2021, Xiamen, China

ACM ISBN 978-1-4503-8432-2/21/10…$15.00
https://doi.org/10.1145/3501409.3501674

## CCS CONCEPTS

• Computing methodologies~Artificial intelligence~Computer vision~Computer vision problems~Object detection

## Keywords

UAV, object detection, YOLO, lightweight, edge computing

## 1 Introduction

When using the deep learning based object detection framework to deal with the problem of UAV, due to the small memory and limited computing power of UAV, the existing recognition algorithms have poor recognition effect and non-ideal processing results on vehicle objects in UAV images. Therefore, it is of great research significance to realize the low-altitude object detection of UAVs by improving the YOLO algorithm.

Object detection methods based on deep learning are mainly divided into two-stage detection and one-stage detection. The two-stage object detection method includes a preprocessing step for proposals. First, the algorithm generates a series of proposals that may contain objects. Then, based on the characteristics of proposals, it performs object classification and bounding box regression, such as R-CNN [1], Fast R-CNN [2], and Faster R-CNN [3]. However, due to the long detection time of the two-stage algorithm, the speed and efficiency of real-time detection cannot be achieved, so a one-stage object detection algorithm appears. The one-stage object detection algorithm directly converts the location and classification problem into a regression problem without extracting proposals. Such methods include SSD (Single Shot Multibox Detector) [4], YOLOv3 (You Only Look Once v3) [5].

YOLO [6] is a one-stage object detection method proposed by Redmon in 2016. It simplifies the entire process of object detection and integrates object determination and recognition. The detection speed can reach 45 frames per second. However, the coarse division of YOLO cells results in a large error in object location regression. Therefore, it is still difficult to deploy on embedded platforms or platforms with small memory capacity.

This paper mainly provides a solution to the problem of low power consumption and small video memory edge computing nodes occupying too much memory when inferring high pixel density images. By transplanting the MobileNetv3 network structure, the computing power requirements on the edge computing platform are reduced. Through experimental comparison of YOLOv4-tiny, YOLOv5s, and YOLOv5-MobileNetv3Small, it is concluded that the optimized YOLOv5-MobileNetv3Small network structure model occupies less memory and is easier to deploy in memory-scarce edge systems.

## 2 YOLO's Network Structure Analysis

YOLOv4 [7] is a one-stage object detection algorithm with strong real-time performance. The algorithm consists of the BackBone network for feature extraction, the Neck for feature fusion, and the detection Head for classification and regression. Compared with the YOLOv3 object detection algorithm, the YOLOv4 algorithm is based on the "Darknet53+FPN+YOLO-Head" algorithm structure of YOLOv3 and integrates the excellent algorithm model ideas and training skills of deep neural networks in recent years. The backbone network is based on Darknet and integrates the idea of CSPNet [8] algorithm to form CSPDarknet, which achieves the effect of reducing the number of network calculations while ensuring the accuracy of the network. Neck is the Path Aggregation Network [9] (PANet) by replacing Feature Pyramid Networks (FPN) to Spatial Pyramid Pooling (SPP) of YOLOv3. It propagates the output of the deep feature by the backbone network to the shallow layer, which improves the problem of shallow feature loss caused by the propagate from shallow features to deep features in FPN. The detection head continues to use the YOLO-Head in YOLOv3. Thus, the model forms a model structure of "CSPDarknet+PAN-SPP+YOLO-Head".

YOLOv5 [10] has four network models: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. And the network structure of YOLOv5 consists of four parts: input, BackBone, Neck, and Prediction.

(1) The input adopts Mosaic data augmentation, adaptive anchor calculation, and adaptive image scaling. Through Mosaic data augmentation, the input images can be spliced by random scaling, random cropping, and random arrangement, which can improve the detection efficiency of small objects.

(2) Focus structure and CSP structure are added to BackBone. The Focus structure of YOLOv5 is utilized to implement crop operations and design two CSP structures.

(3) The Neck structure is the same as that of YOLOv4, i.e., the structure of FPN + PAN-SSP.

(4) The prediction applies GIOU_Loss as the loss function of the Bounding Box.

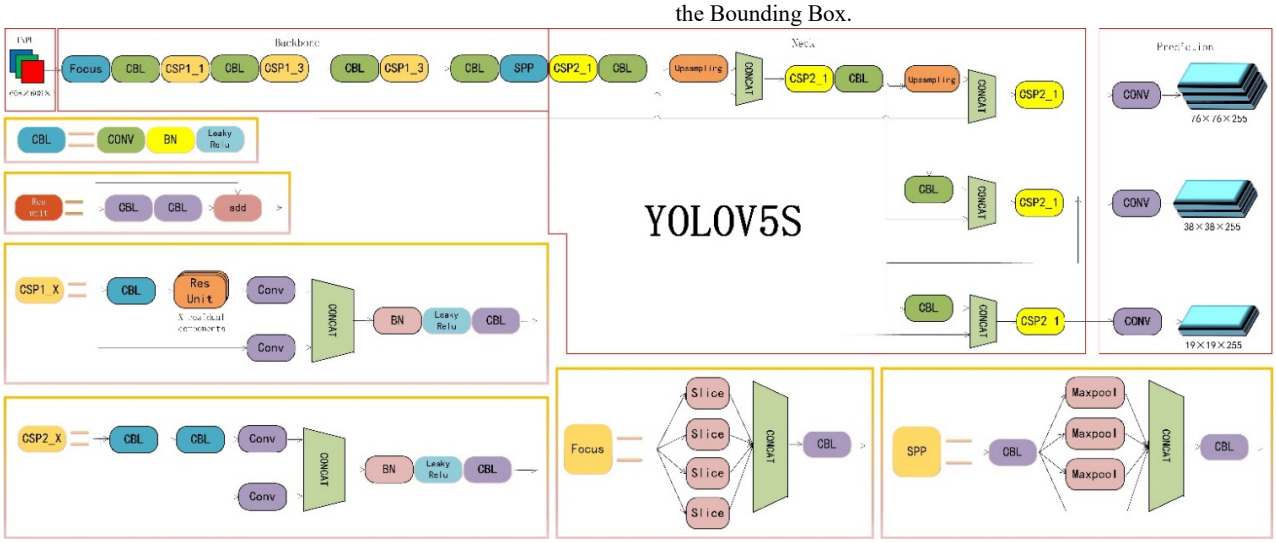


**Fig. 1 The Framework of YOLOv5.**

## 3 Optimal Design of Network Structure

An important part of the object detection algorithm is the BackBone structure of feature extraction. Its complexity largely determines the processing time and detection accuracy of the object detection algorithm. MobileNetv3 [11] is the latest structure of the MobileNet network model based on the depth-wise separable convolution network. The MobileNetv3 network structure is introduced to replace the original backbone feature extraction network in YOLOv5. Two models, Large and Small, are proposed in the MobileNetv3 model structure. Based on the network structure deployment of embedded devices, too large and complex neural network models may face the problem of insufficient memory. Therefore, the Small model is selected as a control. The framework of MobileNetv3Small is shown in **Table 1**.

**Table 1 The framework of MobileNetv3Small**

| Input | Operator | exp size | #out | SE | NL | s |
|---|---|---|---|---|---|---|
| $224^2$ x 3 | conv2d, 3x3 | - | 16 | - | HS | 2 |
| $112^2$ x 24 | bneck, 3x3 | 16 | 16 | √ | RE | 2 |
| $56^2$ x 24 | bneck, 3x3 | 72 | 24 | - | RE | 2 |
| $28^2$ x 24 | bneck, 3x3 | 88 | 24 | - | RE | 1 |
| $28^2$ x 40 | bneck, 5x5 | 96 | 40 | √ | HS | 1 |
| $14^2$ x 40 | bneck, 5x5 | 240 | 40 | √ | HS | 1 |
| $14^2$ x 40 | bneck, 5x5 | 240 | 40 | √ | HS | 1 |
| $14^2$ x 40 | bneck, 5x5 | 120 | 48 | √ | HS | 1 |
| $14^2$ x 48 | bneck, 5x5 | 144 | 48 | √ | HS | 1 |
| $14^2$ x 48 | bneck, 5x5 | 288 | 96 | √ | HS | 2 |
| $7^2$ x 96 | bneck, 5x5 | 576 | 96 | √ | HS | 1 |
| $7^2$ x 96 | bneck, 5x5 | 576 | 96 | √ | HS | 1 |
| $7^2$ x 96 | conv2d, 1x1 | - | 576 | √ | HS | 1 |
| $7^2$ x 576 | Pool, 7x7 | - | - | - | HS | 7 |
| $1^2$ x 576 | conv2d 1x1 | - | 1280 | - | HS | 1 |
| $1^2$ x 1280 | conv2d 1x1 | - | k | - | HS | |

The optimized YOLOv5-MobileNetv3Small network is shown in **Fig. 2**.

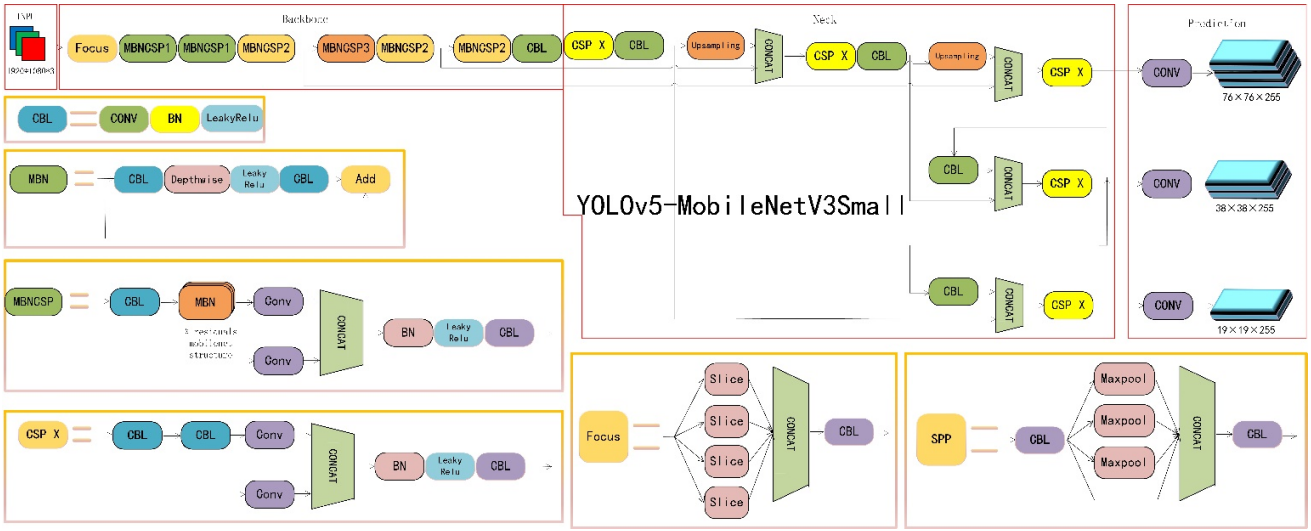


**Fig. 2 The Framework of YOLOv5-MobileNetv3Small**

## 4 Experiments and Analysis

The training and verification platform is based on a Windows10 operating system, equipped with an intel i7-9700F @4.9GHz (8-cores) processor, a graphics memory with 8G NVIDIA GeForce Colorful RTX2060 super, and a computer memory capacity of 32G. The project is developed based on the Conda environment, and the experiment uses the deep learning framework Pytorch for deployment and optimization.

The UAV test platform uses Amu Lab [12] P450 quadrotor UAV. Built-in ROS (Robot Operating System) can realize UAV control and a variety of cutting-edge algorithms. Prometheus is an open-source autonomous UAV software platform developed by Amu Labs, which provides a complete solution set for the intelligent and autonomous flight of UAVs.

All models in this study are trained, tested, and evaluated based on the VisDrone2019 challenge dataset [13]. Visdrone2019 is collected by Tianjin University Machine Learning and Data Mining Lab. The data set is captured by UAV, including 288 video clips, 261908 frames and 10209 still images, and is composed of more than 2.6 million manual annotation boxes of common targets (such as people, cars, bicycles and tricycles). The data set also provides important attributes such as scene visibility, object category, and occlusion to improve the difficulty and algorithm of the target detection.

Since the dataset cannot be directly deployed and trained using the YOLOv5-based model framework, it is necessary to perform corresponding data preprocessing on the dataset to form labels that conform to the model. The dataset needs to convert the original labels provided by VisDrone into acceptable labels in the YOLOv5 before the model can be deployed and trained accordingly.

To evaluate the improved YOLO object detection algorithm, recognized evaluation metrics are used, including precision (P), recall (R), Average Precision (AP), and mean Average Precision (mAP) to measure the detection accuracy, Frames Per Second (FPS) to measure the detection speed, etc.

By passing the classification mark, can get four parameters, correctly classified as a correct case, indicated as True Positive, incorrectly classify the correct example to the negative, indicated as False Negative, correctly classify the negative , Expressed as True Negative, incorrectly classified into a correct example, indicating that False Positive. From these four values, the Precision rate can be obtained. Display Formula without Number 1.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

The Recall rate can be obtained. Display Formula without Number 2.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

The AP value scale has an algorithm to detect the precision and recall rate when a certain type of goal is detected. The larger the AP value, the better the detection effect of the algorithm. Formula without Number 3.

$$mAP = \frac{\sum_{i-1}^{n} AP_i}{n} \tag{3}$$

In the experiment, YOLOv4-tiny and YOLOv5s are used as control algorithms for training and testing, and YOLOv5-MobileNetv3Small is used for model verification. At a resolution of 1920*1080, 300 rounds of training are conducted to obtain the corresponding weight files. The weight file is used to test the accuracy of the Visdrone test data set to obtain the test results, and the performance parameters obtained are shown in **Table 2**.

**Table 2 Performance evaluation metrics and memory usage of different models**

| Model | Precision | Recall | mAP @ 50 | FPS | Memory usage |
|---|---|---|---|---|---|
| YOLOv4-tiny | 0.236 | 0.379 | 0.262 | 8.26 | 398MB |
| YOLOv5s | 0.407 | 0.512 | 0.470 | 6.17 | 1283MB |
| YOLOv5-MobileNetv3Small | 0.509 | 0.438 | 0.461 | 4.56 | 354MB |

By optimizing MobileNetv3Small, whose backbone network is YOLOv5s, although it is not outstanding in terms of speed and detection accuracy, it benefits from a deep wise separable convolution network and a lightweight attention model. YOLOv5-MobileNetv3Small only occupies 354 MB of memory. Compared with the original YOLOv5s, the memory usage is reduced by 72.4%, with almost the same accuracy. Compared with YOLOv4-tiny, the average accuracy is increased by 76%, and the memory usage is smaller. When deployed in embedded devices with small memory or a quadrotor UAV system that uses images for localization, it has a huge performance advantage. The detection result is shown in **Fig. 3**.


**Fig. 3 The Output Image after Testing**

## 5 Conclusion

This study improves the low-altitude real-time object detection method based on YOLOv5. By transplanting the MobileNetv3 network structure, it provides a solution to the problem of low power consumption and small video memory edge computing nodes occupying too much memory when inferring high pixel density images. The experimental results show that using the Visdrone2019-DET dataset to train and test the optimized model can achieve an effective performance improvement. Using the Jetson NX edge computing platform equipped with the UAV to process 1920*1080 video streams, the memory usage of the optimized YOLOv5-MobileNetv3Small network model can be reduced by 72.4%, which has a certain engineering application value.

## Funding project

The “Three Innovations” Excellent Society Construction Project of Fujian Provincial Association for Science and Technology Service (Fujian Science Association [2019] No. 8) and The Natural Science Foundation of Fujian Province (2019J01773, 2018J01640, 2017J01728).

## References

[1] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[2] GIRSHICK R. Fast r-cnn[C]. Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[3] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28: 91-99.

[4] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

[5] REDMON J, FARHADI A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[6] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[7] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[8] WANG C Y, LIAO H Y M, WU Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390-391.

[9] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.

[10] JOCHER G. YOLOv5[EB/OL]. [2020-08-10]. https://github.com/ultralytics/YOLOv5.

[11] HOWARD A, SANDLER M, CHU G, et al. Searching for mobilenetv3[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1314-1324.

[12] https://github.com/amov-lab/Prometheus.git

[13] DU D, ZHU P, WEN L, et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019: 213-226.