M-YOLO: A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3

Shan Huang¹, Ye He¹ and Xiao-an Chen¹*
¹College of automotive engineering, Chongqing University, Chongqing 400044, China
*E-mail address: xachen@cqu.edu.cn

Abstract. Vehicle detection at nighttime plays a vital role in reducing the incidence of night traffic accidents. In order to further improve the accuracy of nighttime vehicle detection, and to be suitable for constrained environments (such as: embedded devices in vehicles), this study proposes a deep neural network model called M-YOLO. First, M-YOLO's feature extraction backbone network used the lightweight network MobileNet v2. Second, the K-means algorithm is reused to cluster the dataset to obtain the anchor boxes which are suitable for this paper. Third, M-YOLO uses the EIoU loss function to continuously optimize the model. The experiments showed that the average precision (AP) of proposed M-YOLO can reach 94.96%. And ten frames per second (FPS) were processed in a constrained environment. Compared with YOLO v3, the proposed model performs better in detection accuracy and real-time performance.

1. Introduction

In recent years, with the increase of vehicles, the incidence of traffic accidents is getting higher and higher. It not only seriously threatens people's lives, but also causes huge economic losses. If the computer system can be used to perceive the surrounding environment in time during driving, discover possible dangers as early as possible, then driving safety could be greatly improved. Vehicle detection, as the basis and premise task of intelligent driving technology, has attracted wide attention from scholars. Especially in the nighttime environment where the brightness and contrast are weak and the probability of traffic accidents is high, vehicle detection is extremely important.

Existing vehicle detection algorithms are mainly divided into two types: based on traditional feature extraction and based on deep learning. Vehicle detection methods based on traditional methods use low-level features for feature extraction [1], especially the car light information in the night. Hsia used the Hough transform to detect the lane lines, and after obtaining the light source data of each lane, the vehicle count was realized according to the aspect ratio and spacing of the light source [2]. Jong proposed an improved multi-scale retina (MSR) image enhancement algorithm, and used a two-stage classifier based on BoF and CNN to reduce false positives [3]. The above night vehicle detection method based on car light features and classifiers is easily affected by road scenes. Due to it rely on manual feature extraction.

The development of deep learning has opened up new world for target detection and greatly improved the accuracy of target detection. Faster-RCNN, YOLO, SSD and other target detection models are used in vehicle detection. Wang used five deep learning target detection models on the KITTI dataset to test the effectiveness of deep learning in vehicle detection [4]. He et combined with millimeter-wave radar to locate the vehicle position, and achieved rapid detection of vehicle targets by optimizing the network structure of YOLOv2. Which further shortened the detection time [5]. However, these algorithms have a huge amount of calculation and require powerful hardware facilities to work.
Therefore, this paper proposed M-YOLO. It can be used in constrained environments, suitable for application in embedded devices of vehicles. And it has a higher detection rate.

2. M-YOLO architecture
This paper aims to explore a more suitable detection algorithm used in the vehicle's embedded device. Proposed M-YOLO chose the lightweight network Mobilenet v2 [6]as the backbone network for feature extraction. Mobilenet v2 can greatly reduce the parameters and calculations of the model, and improve the running speed of the model. The detection head still uses the multi-scale prediction head of YOLO v3[7]. The loss function is optimized using the EIoU loss function. In addition for the dataset in this article, the k-means algorithm is used to re-cluster, and get Suitable anchor boxes. The network structure of M-YOLO proposed in this paper is shown in Figure 1.

2.1. Backbone networks
Generally speaking, the easiest way to compress a model is to reduce the number and size of the model and use a shallow neural network. However, the detection efficiency of shallow networks is generally much lower than that of deep neural network models. Therefore, this article chooses the method of redesigning the network model to compress the network model. That is, using the MobileNet v2 network as the feature extraction network of M-YOLO. The MobileNet v2 network structure is shown in Table 1. Lightweight network MobileNet v2 can reduce the computational complexity of parameters and models. The bottleneck layer in the above table is shown in the figure 2.

| Input     | Operator     | t | c | n | s |
|-----------|--------------|---|---|---|---|
| 224²x3    | conv2d       | - | 32| 1 | 2 |
| 112²x32   | bottleneck   | 1 | 16| 1 | 1 |
| 112²x16   | bottleneck   | 6 | 24| 2 | 2 |
| 56²x24    | bottleneck   | 6 | 32| 3 | 2 |
| 28²x32    | bottleneck   | 6 | 64| 4 | 2 |
| 14²x64    | bottleneck   | 6 | 96| 3 | 1 |
| 14²x96    | bottleneck   | 6 | 160| 3 | 2 |
| 7²x160    | bottleneck   | 6 | 320| 1 | 1 |
| 7²x320    | conv2d 1×1  | - | 1280| 1 | 1 |
| 7²x1280   | avgpool 7×7  | - | - | 1 | - |
| 1×1×1280  | conv2d 1×1  | - | k | - | - |

Figure 1. Network structure diagram of M-YOLO

Table 1. Overall structure of MobileNetV2
2.2. Anchors boxes
The author of YOLO v3 used the k-means clustering method on the COCO dataset, selecting 9 cluster centers and dividing them into three groups to predict three types of targets: large, medium, and small. But the COCO data set has a total of 80 categories, covering from small objects such as: toothbrushes and mice to large objects: such as sofas, elephants, and trains. However, this article only detects vehicles and is divided into two categories: vehicles and non-vehicles. Therefore, the anchor boxes clustered according to the COCO dataset is not completely applicable to this article.

The k-means algorithm is a typical clustering algorithm, which obtains the optimal K cluster centers by minimizing the distance between the sample and the cluster center [8]. This article uses the k-means algorithm to re-cluster the dataset used in this article to obtain a new anchor boxes. The resulting anchor boxes are: (30,25), (42,34), (51,37), (54,43), (62,50), (72,49), (79,59), (91,69), (120,85).

2.3. Loss function
The loss function of the deep neural network is used to measure the difference between the predicted value and the true value. The loss function of YOLO v3 is mainly composed of location loss, confidence loss and class loss. In the loss function of M-YOLO, class loss and confidence loss are the same as YOLO v3. But the location loss used EIoU loss.

The confidence loss is:
\[
L_{\text{conf}} = -\lambda_{\text{obj}} \sum_{i=0}^{N} \sum_{j=0}^{G} I_{y_{ij}}^{\text{obj}} \left[ \log(C_i) + (1-C_i) \log(1-C_i) \right] - \lambda_{\text{noobj}} \sum_{i=0}^{N} \sum_{j=0}^{G} I_{y_{ij}}^{\text{noobj}} \left[ \log(1-C_i) + (1-C_i) \log(1-C_i) \right]
\]

The class loss is:
\[
L_{\text{class}} = -\sum_{i=0}^{N} \sum_{j=0}^{G} I_{y_{ij}}^{\text{obj}} \sum_{c_{\text{classes}}} \left[ \hat{p}_c \log(p_c) + (1-\hat{p}_c) \log(p_c) \right]
\]

The location loss of YOLO v3 uses the mean square loss function (MSE) based on Euclidean distance. Compared with Euclidean distance, the IoU loss function [9] can better represent the overlap between the predicted bounding boxes and the true bounding boxes. Therefore, M-YOLO uses EIoU[10] loss function that can truly reflect the width and height of the bounding box.

EIoU is as follows:
\[
L_{\text{EIoU}} = 1 - \frac{|B \cap B_{gt}|}{|B \cup B_{gt}|} + \frac{\rho^2(h, b_{gt})}{c^2} + \frac{\rho^2(w, w_{gt})}{c_w^2} + \frac{\rho^2(h, h_{gt})}{c_h^2}
\]

The total loss function of M_YOLO is as follows:
\[
L_{\text{total}} = L_{\text{bbox}} + L_{\text{conf}} + L_{\text{class}}
\]

3. Experiment
This article uses the public data set provided by Long et al. of Sun Yat-Sen University[11]. The dataset provides 5576 nighttime traffic scene pictures with a resolution of 640×360. During the model training,
90% of the image were randomly selected as the training set, and the remaining 10% were used as the test set. The proposed algorithm trained and tested in a poor environment (a thin and light notebook computer), details as follows: i7-4710MQ CPU, NVIDIA GeForce GTX 860M.

3.1. Model size and BFLOP
Table 2 compares the size, BFLOP and network parameters of the models from YOLO v1 to YOLO v3. In order to ensure the effectiveness of comparison, all input images adopt $416 \times 416$ pixel size.

| Model     | Size   | Parameter | BFLOP |
|-----------|--------|-----------|-------|
| YOLO v1   | 1.1G   | 271.6M    | 40.19 |
| YOLO v2   | 194MB  | 50.5M     | 62.94 |
| YOLO v3   | 237MB  | 61.9M     | 65.86 |
| M-YOLO    | 71.8MB | 18.7M     | 36.00 |

It can be seen from the Table 2. That model size, Parameters and BFLOP of the M-YOLO model proposed in this paper is much smaller than YOLO v1 to v3. It means that the complexity of the model is much lower.

3.2. Ablation experiment
In order to verify the effectiveness of the proposed network M-YOLO, an ablation experiment was carried out in this paper. Table 3 compares the AP of reselecting the anchor boxes, replacing the backbone network with MobileNet v2, and different loss functions on the basis of YOLO v3. Meanwhile, Table 3 compares FPS of YOLO v3 and M-YOLO.

| Model     | Anchor Boxes | MobileNet v2 | loss | AP, % | FPS |
|-----------|--------------|--------------|------|-------|-----|
| YOLO v3   | √            | MSE          | MSE  | 76.51 | 3   |
| M-YOLO    | √            | √            | DIoU | 88.57 | 10  |

Compared with YOLO v3, when the anchor boxes are replaced, the average accuracy of vehicle detection has been greatly improved. AP increased from 76.51% to 87.33%. Secondly, after the backbone network was replaced with MobileNet v2 from DarkNet53, the AP was slightly reduced, but the reduction was very small. Finally, the DIoU loss function, CIoU loss function, and EIoU loss function are used as the location loss function to train the M-YOLO network. The average precision is gradually increased. The final average accuracy of M-YOLO is 94.96%. Moreover, the AP of M-YOLO is 18.45% higher than YOLO v3 but three times faster than YOLO v3.

3.3. Visualization of image results of specific experiments
Figure 3 shows the results of the test in this paper. Figures (a) to (c) show the detection of multiple vehicles in the case of a large traffic volume in an urban environment. Figures (d) to (f) show the effect of vehicle detection when the vehicle is occluded. In various scenes, M-YOLO achieve better detection.

4. Conclusion
This paper proposes a deep neural network model M-YOLO based on YOLO v3. M-YOLO's feature extraction backbone network uses the MobileNet v2 network. The depth separable convolution proposed by the MobileNet network can greatly reduce the amount of parameters and the size of the model. The detection head still draws on the multi-scale target prediction mechanism of YOLO v3. At the same time,
the K-means algorithm is used to re-cluster the dataset in this paper to obtain the anchor boxes suitable for this paper. In addition, M-YOLO uses the EIoU loss function to continuously optimize the model. The results show that the average accuracy rate of M-YOLO can reach 94.96%, and in poor environment the frame rate per second can reach ten. Moreover, the AP of M-YOLO is 18.45% higher than YOLO v3 but three times faster than YOLO v3.

![Figure 3. Test results](image_url)

Acknowledgments
This research was supported by Chongqing Technology Innovation and application demonstration special industry key research and development project "new energy commercial vehicle new intelligent and efficient powertrain key technology research and development and industrial application" (project number cstc2018jszx-cyzdX0069).

Reference
[1] Wang, X., Zhao, X.K. (2016) Vision-based two-step brake detection method for vehicle collision avoidance. Neurocomputing, 173: 450-461.
[2] Hsia C H, Yen S C, Jang J H. (2019) An Intelligent IoT-based Vision System for Nighttime Vehicle Detection and Energy Saving. Sensors and materials, 31:1803-1814.
[3] Jong C C, Yong-Sheng C, Jiann-Der L. (2017) Improving Night Time Driving Safety Using Vision-Based Classification Techniques. Sensors, 17: 139-146.
[4] Wang, H. (2019) A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection. IEEE Intelligent Transportation Systems Magazine, 11: 82-95.
[5] He Y, Li L. (2018) A Novel Multi-source Vehicle Detection Algorithm based on Deep Learning. In: 2018 14th IEEE International Conference on Signal Processing (ICSP). Bei Jing.
[6] Sandler M, Howard A, Zhu M. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake. pp.4510-4520.
[7] Redmon J, Farhadi A. (2018) YOLOv3: An Incremental Improvement. arXiv e-prints.
[8] Arthur D, Vassilvitskii S. (2007) K-Means++: The Advantages of Careful Seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Louisiana.
[9] Rezatofighi, S. H. (2019) Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In: CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach. pp.658-666.
[10] Zhang Y F, Ren W, Zhang Z. (2021) Focal and Efficient IOU Loss for Accurate Bounding Box
Regression. ArXiv. abs/2101.08158.

[11] Sun Yat-sen University. (2015). http://www.carlib.net.