Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/52/20241500

Vehicle detection in complex scenarios based on YOLOv5

Xingyue Lin[1], Zhaoting Zhu[2,3,4]

[1]School of Games and Animation, China Academy of Art, Hangzhou, Zhejiang, 310000, China
[2]School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, Chongqing, 400000, China

[3]2021214269@stu.cqupt.edu.cn
[4]Corresponding author

Abstract. With the acceleration of urbanization, traffic congestion, accident risk, and other issues have become increasingly prominent, so vehicle recognition and optimization technology has become particularly important. Based on the YOLOv5 deep learning model, this study proposes a vehicle recognition and optimization method to address key issues in urban traffic management and intelligent driving. This study first tested YOLOv5 and trained and optimized the YOLOv5 model to enhance the model's ability to detect vehicles in various complex scenarios, such as rainy days and the coexistence of people and vehicles, which have a large number of interference factors. Through experimental verification, our improved model improves the accuracy of detection in such scenarios in vehicle recognition tasks, and finds the deficiencies in the experiment and proposes corresponding solutions.

Keywords: Vehicle detection, YOLOv5, Complex scenarios.

1. Introduction

With the accelerating process of urbanization and the sharp increase in traffic flow, urban traffic management is facing unprecedented challenges. Traffic congestion, traffic accidents, and exhaust emissions have become the norm of urban life, which not only brings many inconveniences to residents' travel, but also has a serious impact on the environment and resources. In this context, intelligent traffic management and intelligent driving technology have become crucial to improve traffic efficiency, reduce accident risks, reduce exhaust emissions, and achieve sustainable development of urban transportation.

With the development of artificial intelligence technology, vehicle recognition methods based on the YOLO (You Only Look Once) series detection model have attracted increasing research attention in recent years. As a newer version of the YOLO series model, YOLOv5 has significantly improved both the detection speed and accuracy of general objects. Specifically [1], YOLOv5 is designed for real-time target detection, which means it can detect multiple objects in images or videos at real-time or near real-time speed [2]. The speed of YOLOv5 is usually faster than some traditional target detection algorithms, while it also achieves a good balance between accuracy and speed at the same time. YOLOv5 has achieved excellent performance on multiple target detection datasets and can detect targets of different sizes, which makes it perform well in various scenarios. Therefore, its high

DOI: 10.54254/2755-2721/52/20241500

performance, multi-category detection function, and multi-scale detection function make YOLOv5 a powerful target detection model, and its speed and performance make it an ideal choice for many computer vision applications [3]. It is widely used in object detection, tracking, and classification.

Currently, the academic community has high expectations for using YOLO for vehicle recognition. Researchers are constantly working to change the backbone network of the YOLO model, such as incorporating EfficientNet, to further improve performance. At the same time, the academic community has also conducted research on data augmentation methods to increase the robustness of the YOLO model. Some research focuses on combining visual information with other sensor data such as lidar or radar data to improve the performance and reliability of vehicle recognition.

Though previous works have made significant progress, there are still some challenges in using YOLOv5 to identify vehicles. For example, in today's complex urban backgrounds, vehicles often appear in different environments, such as city streets and highways [4]. These complex backgrounds may lead to false detection or missed detection, which affects accuracy. In some cases, vehicles may be obscured by other vehicles, buildings, or obstacles, which increases the difficulty of detection and recognition. Some visible vehicles also need to be correctly detected and classified [5]. Vehicle recognition tasks often involve multiple vehicle types, including cars, trucks, bicycles, etc. Each type of vehicle has different appearances and characteristics [6], requiring the model to be able to distinguish between them. In some applications, such as autonomous driving, real-time performance is highly required. Therefore, the model needs to maintain high accuracy while running quickly. In some scenarios, the types of vehicles that appear may be imbalanced, which can lead to less-than-ideal recognition performance in this scenario.

To alleviate the above issues and improve the accuracy of vehicle detection in complex scenarios, in this study, we trained the vehicle detection model based on the YOLOv5s model and conducted extensive experiments to analyze its performance for various complex scenarios. Focusing on the above aspects, we will detail the methods in Section 2 and report the detection results in Section 3. Then, we discuss the existing challenges and look forward to the future development direction, which we suppose may bring some new insights to this promising topic [7].

2. Method

2.1. Revisiting YOLOv5 model

YOLOv5 is an object detection algorithm based on deep learning, which is one of the YOLO series of algorithms. It has a great advantage in speed operating speed. Therefore, YOLO can be used to detect objects in real time, that is, to quickly and accurately identify and locate objects in images or videos. According to the depth of the neural network, YOLOv5can be divided into YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x and other versions, which share the similar network structure. In this work, we adopt the version of YOLOv5s as baseline, and the network structure of YOLOv5s algorithm is shown in Fig 1.

As shown in Fig 1, the structure of YOLOv5s algorithm can be divided into Backbone, Neck and Output. The Backbone part is used to extract features from input images. YOLOv5s uses a backbone network architecture called CSPDarknet53. As a lightweight feature extraction network consisting of a series of convolutional layers and residual blocks, CSPDarknet53 has a good capability to represent features and is optimized in terms of reducing parameters and computational complexity. The Neck part uses a structure called BiFPN (Bi-directional Feature Pyramid Network) in YOLOv5s. BiFPN transfers and fuses information between feature maps at different levels through top-down and bottom-up paths to obtain multi-scale semantic information. This structure allows YOLOv5s to detect targets at different scales, which can detect small and large targets and improve detection performance. The Output part consists of a series of convolution layers and full connection layers, which are usually called detection heads. These detection heads can receive multi-scale features from the feature pyramid network, and map them to the prediction box and category probability space through convolution operation.

DOI: 10.54254/2755-2721/52/20241500

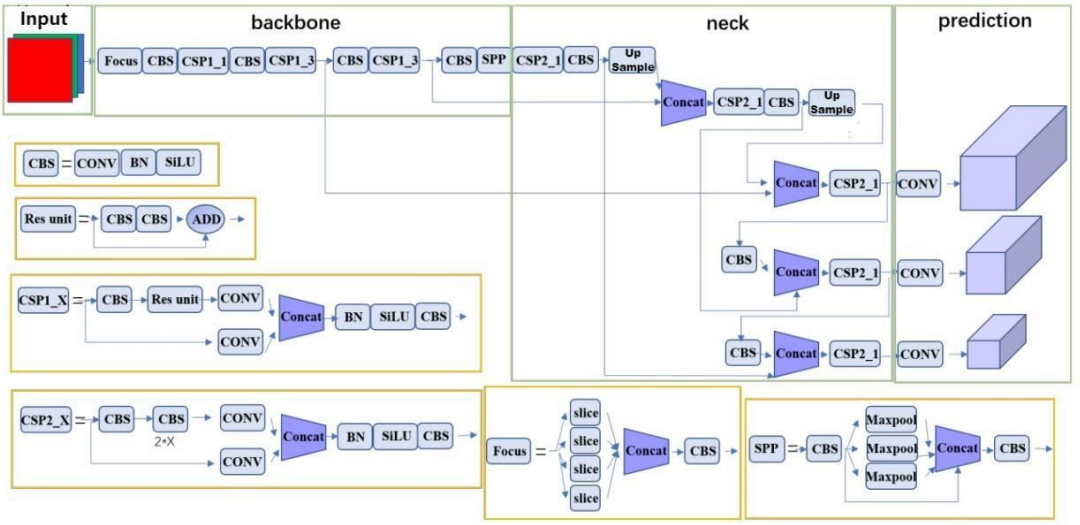

Figure 1. The network structure of YOLOv5s algorithm[8]

2.2. Loss function

The loss function of the YOLO model is optimized by a combination function combining cross entropy loss and mean square error loss. The cross-entropy loss mainly focuses on the identification of target categories. By minimizing the difference between the predicted category and the actual category, the YOLO model focuses more on the accurate classification of targets. The mean square error loss focuses on the positioning of the target position. By minimizing the mean square error between the predicted position and the actual position, the YOLO model realizes the accurate positioning of the target position. The loss fuction is formulated in Equation (1) and (2).

$$\text{Cross-Entropy Loss: } P_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{1}$$

$$\text{Mean Square Error: } L_{CE} = -\sum_{i=1}^{k} y_i \log(P_i) \tag{2}$$

Using these two loss functions synthetically, the model can not only accurately identify the target category, but also accurately locate the target position in the target detection task. Cross entropy loss provides clear target classification information for the model, which enables the model to quickly learn target classification information. At the same time, the mean square error loss has made more detailed adjustments to the location of the bounding box, which can further improve the accuracy of target positioning. This strategy of comprehensive use of loss function effectively improves the performance of the model in the target detection task.

2.3. Original data and preprocessing

To complete the vehicle detection task, we chose COCO (Common Objects in Context)as our training dataset. COCO provides a bounding box for each object in the object detection task. Moreover, the images in the COCO cover a variety of scenes, from indoor to outdoor, from natural landscapes to urban environments. Therefore, COCO enables the model to have good detection ability in various backgrounds and environments [9-10], especially in vehicle recognition tasks. For vehicle detection tasks, we filter out specific categories and scenes from the dataset, including "vehicles" and "pedestrians", and extract images from various scenes to build a training set. In addition, in order to adapt the model to different environments and scenes, images of various scenes such as city streets, highways, parking lots, etc. were selected from the COCO dataset and merged with the images of the

DOI: 10.54254/2755-2721/52/20241500

training set. Besides, we built a test set to evaluate the performance of the model, ensuring that the distribution of categories and scenarios in the test set was similar to the training set.

In the data pretreatment stage, all images are adjusted to the same size to ensure the consistency of the input model, and normalize the image to scale the pixel value between 0 and 1, so as to better input the neural network model. Meanwhile, data enhancement techniques, such as random clipping, flipping, rotation, etc. are applied to expand the training set and enhance the generalization ability of the model. Also, the COCO's annotation format is converted into the format required by the model.

2.4. model training

In the experiment, a training script is created to train the YOLOv5 model, enabling the vehicle recognition model to have better performance and cope with complex situations. The COCO is chosen as the training dataset, because COCO contains a large number of different categories of items and a variety of different environments, so the model can perform effective recognition detection in various situations [11].

The training process consists of several steps. The first step is pre-training, which includes training parameters, number of cycles, number of training rounds, etc. At the same time, some variables are initialized to save some data, such as the best weight and the last weight. The second step is model initialization. In this step, a YOLOv5 model is created, loaded, and trained with its weights. After that, the data is loaded and preprocessed, which includes loading the dataset and converting the image into tensors, normalization operations, etc. The step of adjusting the optimizer consists of two parts. The first is to select the optimizer (SGD/Adam), and the second is to set up a learning rate scheduler to dynamically adjust the learning rate. Each training cycle includes the necessary steps of data loading and preparation, forward propagation, loss calculation, backpropagation, model validation, etc. After all epochs are completed, the training time and other information are recorded.

2.5. Experiment Setting

When training the YOLOv5 model, we utilized PyTorch as the deep learning framework and employed a workstation equipped with an NVIDIA GPU (with a memory of 16GB) for accelerated computations. We set the initial learning rate to 0.01 and implemented a dynamic adjustment strategy, reducing the learning rate to 10/1 of its current value every 30 epochs. A batch size of 16 was chosen to fully use the GPU resources. The training lasted for 100 epochs, with random data augmentation technique such as horizontal flipping, random cropping and so on to enhance the model performance. The input image size was set to 640*640, ensuring effective detection at higher resolutions. We employed the SGD optimizer with a momentum of 0.9 to control model complexity. Finally, a multi-scale training strategy was implemented, selecting training image sizes randomly to enhance the model's ability to detect objects in various scales.

3. Experimental results and analysis

In this section, we presented and analyzed the results. In order to analyze the model more accurately and rigorously, we compared the confidence of vehicles recognized in different occasions, such as rainy days and the coexistence of people and vehicles, before and after model training and the number of recognized vehicles and obtained the results, as shown in Table 1, Table 2 and Figure 2.

DOI: 10.54254/2755-2721/52/20241500

3.1. Comparison and analysis of recognition results of rainy days

Table 1. Detecting result for the scenario of rainy days

Index / Ref	Confidence (Before Training)	Confidence (After Training)
Car1	0.90	0.92
Car2	0.87	0.83
Car3	0.85	0.79
Car4	0.79	0.73
Car5	0.76	0.73
Car6	0.72	0.72
Car7	0.61	0.69
Car8	Unrecognized	0.63
Car9	Unrecognized	0.51

In rainy days, vehicle recognition work and recognition equipment are usually affected by the environment, such as a series of interference such as rain covering the camera's view, reflection of rain on the road. These interferences can easily lead to poor recognition results. From Table 1, we can see that the number of vehicles before training is significantly lower than the results after training. At the same time, the confidence of Car1 and Car7 increased by 0.02 and 0.08 respectively. After training, the model can recognize things that could not be recognized before such as Car8 and Car9. It can be seen that the recognition performance has been significantly improved.

3.2. Comparison and analysis of recognition results of coexistence between people and vehicles

Table 2. Detecting results for the scenario when vehicles and people coexist

Index / Ref	Confidence (Before Training)	Confidence (After Training)
Car1	0.75	0.88
Car2	0.72	0.75
Person1	0.86	0.87
Person2	0.84	0.82
Person3	0.79	0.81
Person4	Unrecognized	0.76
Person5	Unrecognized	0.48
Person6	Unrecognized	0.42
Motorbike1	Unrecognized	0.69
Motorbike2	Unrecognized	0.56
Motorbike3	Unrecognized	0.55
Bicycle1	Unrecognized	0.66
Bicycle2	Unrecognized	0.32
Dog1	0.92	Unrecognized
Dog2	0.87	Unrecognized
Dog3	0.84	Unrecognized

DOI: 10.54254/2755-2721/52/20241500

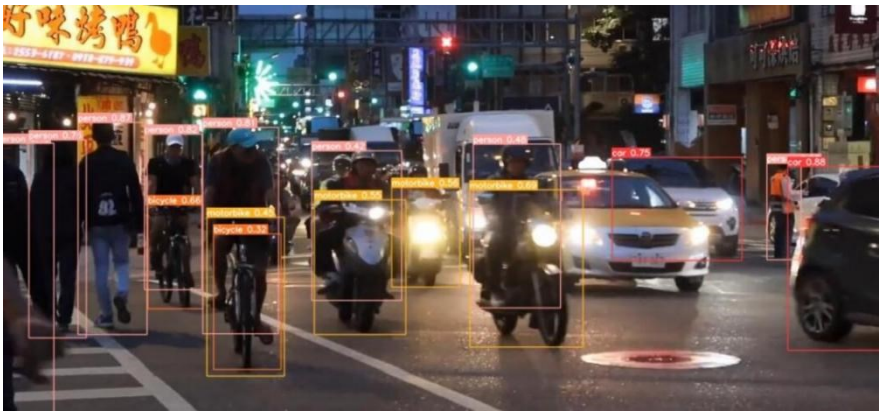

Figure 2. Visualization of detection result of our method

Vehicle recognition will also encounter many challenges when people and vehicles coexist. Urban traffic environments are usually very complex, including roads, sidewalks, intersections, etc. In such an environment, vehicles may appear from all directions and angles, making accurate identification more difficult. At the same time, the wide variety of vehicles, like small cars, large trucks, bicycles and motorcycles, etc., they have different sizes, shapes and colors. So the model has to have powerful generalization capabilities to recognize all types of vehicles. From Table 2, we find that the recognition effect is not good, and there are false detections, such as Dog1 to Dog3 in Table 2. After training, the model can not only improve the false detection problem, but also can identify more types of vehicles, such as Motorbike and Bicycle in Table 2, and except for Person2, the confidence rates of other recognition objects have been improved to varying degrees.

4. Discussion

4.1. Confidence rate fluctuations

From the result above, we noticed that the confidence rate before and after model training fluctuated to a certain extent in the case of recognition on rainy days. For example, the confidence rate of Car2 decreased by 0.04 and the confidence rate of Car3 decreased by 0.06, so we investigated the related reasons. We found that the COCO dataset mainly contains images under sunny conditions, and there are large visual differences between rainy and sunny scenes. This means that the model may not have enough training data to accurately learn vehicle characteristics in rainy weather scenarios. This may lead to model performance degradation in this situation. There are obvious differences in image characteristics between rainy scenes and sunny scenes, and more detailed preprocessing may be needed to mitigate this effect. For example, trying to apply some enhancement techniques to the training data, such as blurring, contrast adjustment, etc., to simulate rainy scenes.

4.2. Poor distant target recognition effect

When identifying distant targets, we found that the recognition results of the trained model were actually not ideal. The model's recognition of nearby targets was relatively good, but there were misdetections or even missed detections of distant targets. In this case, we noticed that the YOLO series models are trained at a single scale, which means that the model will have better recognition results for targets close to the training scale during the prediction process, but will have poor recognition results at distant locations. Therefore, some model hyperparameters need to be adjusted, such as the size of the Anchor Box, to help it adapt to distant targets. Therefore, in subsequent improvement research, methods such as data enhancement can be used to further optimize the model to obtain better recognition results.

DOI: 10.54254/2755-2721/52/20241500

5. Conclusion

This article proposes a vehicle recognition algorithm based on the YOLOv5 algorithm. In order to solve the problem that the original model cannot accurately identify vehicles in rainy weather and that the original model misses or even misjudges the target when people and vehicles coexist, the model is continuously trained using the COCO data set and the SGD/Adam algorithm is integrated into the YOLOv5 algorithm. In the method, a learning rate scheduler is set to dynamically adjust the learning rate to improve the vehicle recognition ability of the model in rainy days or in environments where humans and vehicles coexist. Experimental results show that the model has a high accuracy in identifying targets, which is 52% higher than the accuracy of the original model. The performance of the model has also been greatly improved, and the accuracy in video detection is also high, and can be used for real-time vehicle detection. detection. This article is limited to a small number of experimental samples, and the confidence of the trained model fluctuates to a certain extent. In the future, the performance of the model will be further optimized from the perspectives of data enhancement and network optimization, and vehicle recognition methods in various environments will continue to be studied based on the algorithm in this article.

Authors contribution

All authors contributed equally to this research, and their names are listed in alphabetical order.

References

[1] Huang C L , Liao W C .A vision-based vehicle identification system[C]//International Conference on Pattern Recognition.IEEE, 2004.DOI:10.1109/ICPR.2004.1333778.

[2] Ferryman J M , Worrall A D , Sullivan G D ,et al.A Generic Deformable Model for Vehicle Recognition[J].BMVA Press, 1995.DOI:10.5244/C.9.13.

[3] Zheng W .Object Detection for Construction Waste Based on an Improved YOLOv5 Model[J].Sustainability, 2022, 15.DOI:10.3390/su15010681.

[4] Zhi-Hong Z .Lane Detection and Car Tracking on the Highway[J].Acta Automatica Sinica, 2003, 29(3):p.450-456.

[5] Hwang B G .Car teaching system for Recognize of Drive Way of Vehicle:KR20160096779[P].KR101812156B1[2023-10-15].

[6] Simonyan K , Zisserman A .Very Deep Convolutional Networks for Large-Scale Image Recognition[J].Computer Science, 2014.DOI:10.48550/arXiv.1409.1556.

[7] Shen S R , Zhang X , Yan W ,et al.An improved UAV target detection algorithm based on ASFF-YOLOv5s.[J].Mathematical biosciences and engineering : MBE, 2023, 20 6:, 10773-10789. DOI:10.3934/mbe.2023478.

[8] Fang Y, Guo X, Chen K, et al. Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model[J]. BioResources, 2021, 16(3): 5390.

[9] Doan A , Okatan A , Etinkaya A .Vehicle Classification and Tracking Using Convolutional Neural Network Based on Darknet Yolo with Coco Dataset[C]//ICAIBDEA 2021 International Conference on AI and Big Data in Engineering Applications.2021.

[10] Robalinho M .Learned lessons with city traffic tests using real time object detection with tensorflow and COCO dataset model[J]. 2019

[11] Bishop C M , Bishop C , Bishop C M ,et al.Neural Network for Pattern Recognition[J]. 1995.