Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Vehicle detection in foggy weather based on an enhanced YOLO method

To cite this article: Wei Li 2022 *J. Phys.: Conf. Ser.* **2284** 012015

Vehicle detection in foggy weather based on an enhanced YOLO method

Wei Li[1*]

[1] Computer Science, Xidian University, Xi'an, Shaanxi, 710126, China

[*]Corresponding author's e-mail: wli_9@stu.xidian.edu.cn

Abstract. Vehicle detection is the key to driverless technology. For safety, driverless technology requires extremely high accuracy and real-time for vehicle detection in different situations. In this paper, we study an enhanced YOLO -based algorithm for vehicle detection in foggy weather conditions. We add a dehazing module in the YOLO model for more information restoration, which is built by the multi-scale retinex with color restoration (MSRCR). And the enhanced model is trained with the augmentation data processed MSRCR for more stable performance. We evaluate our method in the public dataset, the results show the enhanced YOLO model has better performance than conventional YOLO in vehicle detection in foggy weather.

1. Introduction

Driverless technology is one of the hottest technologies today. Improving the accuracy of vehicle detection technology can enhance the safety of driverless technology while helping to make judgments about whether a car on the road is in violation of the law. Vehicle detection technology is one of the keys to driverless technology. The current vehicle detection technology is mainly a camera-only vision scheme and a LIDAR perception scheme. The image sensor in the vision scheme is able to acquire complex environmental information of the surroundings at a high frame rate and high resolution, and is inexpensive, which is the method chosen by Tesla.

At present, there are two types of vision-based vehicle detection: traditional machine vision ways and deep-learning-based methods. The former separates the vehicle from a fixed background image by its motion. The method can be divided into three different categories: continuous video frame difference method [1], optical flow method [2], background subtraction method [3].

In terms of deep learning for vision solutions, object detection based on convolutional neural networks (CNN [4][5]) has become mainstream. In the two-stage detection method, candidate boxes of objects are generated by various algorithms, and then objects are classified by convolutional neural network.

Region-CNN (R-CNN) [6] performs selective region search in images based on color, texture, edges, etc. However, the image that is input to the convolutional network should be of a fixed size, and each frame of the candidate frame needs to be extracted by CNN + SVM classification, which requires a large amount of calculation, resulting in very slow detection by R-CNN.SPP-NET [7] combines the spatial pyramid method to realize multi-scale input of CNNs, and joins the ROI pooling layer, which allows the network to input images of different kinds of sizes with a fixed output. In the one-stage methods, the main ones are Single Shot Multibox Detector (SSD) [8]and You Only Look Once (YOLO) [9]. Although the above method works well in normal weather, it is not accurate enough in foggy weather. In recent years, scholars have done some innovative research on vehicle detection in foggy weather. Rahul [10] uses Adaptive Gaussian Thresholding Technique to make the image sharper and incorporates a low-cost LiDAR to prevent accidental collisions.

Kun [11] uses Complementary Lidar and Radar Signals for multi-modal vehicle recognition in foggy weather. They present Multimodal Vehicle Detection Network (MVDNet). This is a two-stage deep fusion detector, which fuses regional features between multimodal sensor streams by combining content obtained from two sensors to improve vehicle detection in foggy weather.

Both of them use a combination of multiple sensors to improve vehicle detection in foggy weather, which requires high costs. Therefore, this paper hopes to improve the vehicle detection effect in foggy weather by using only the data obtained by the camera in a computer vision way.

In this paper, we try to use a generic road dataset to train YOLO models to achieve vehicle detection in foggy weather and harsh environment. For the actual situation of foggy weather, MSRCR algorithm preprocessing is used to improve the performance.

2. Yolo

You Only Look Once, or YOLO, is a state-of-the-art, real-time object detection system first proposed by Joe Redmon in 2016. Just like its name, you only need to input the image and go through an inference to get the positions of all objects in the image, their categories, and corresponding confidence probabilities. Previous detection systems applied models to images at multiple locations and scales. High scoring areas of the image are considered detections. YOLO uses a completely different method. YOLO applies a single neural network to a full image, divides it into regions, and predicts bounding boxes and probabilities for each region.

The information of each picture is the N objects it contains, and each object has five pieces of information, namely the center position (x, y) of the object, its height (h) and width (w), and its category.

YOLO divides the input image into S x S grids, and each grid is responsible for detecting objects in that grid. If the coordinates of the center position of an object fall into a grid, then the grid is responsible for detecting the object.

YOLOv4 [12] mainly consists of three parts: Backbone, Neck, and Head.

The backbone network used is CSPDarknet53. CSPDarknet53 is based on the Yolov3 backbone network Darknet53, drawing on the experience of CSPNet in 2019, including 5 CSP modules. The CSP module consists of the CBM component and the X Res unint module Concate. CBM is the smallest component in the YOLOv4 network structure and consists of Conv+Bn+Mish activation functions.

Neck is a feature enhancement module, which is mainly composed of CBL components, SPP module, and FPN+PAN method. The FPN layer transfers strong semantic features from top to bottom, while PAN transfers strong localization features from bottom to top. Together, they aggregate parameters from different backbone layers into different detection layers, thereby accelerating the fusion of features at different scales and further improving the capability of feature extraction.

Head uses the obtained features to make predictions, which is a decoding process. In the feature utilization part, YoloV4 extracts multi-scale features for object detection. A total of three feature layers are extracted, which are located in the middle layer, the lower middle layer, and the bottom layer. The shapes of the three feature layers are (19, 19, 255), (38, 38, 255), (76,76,255) respectively. These three feature maps are the detection results of the entire YOLO output.

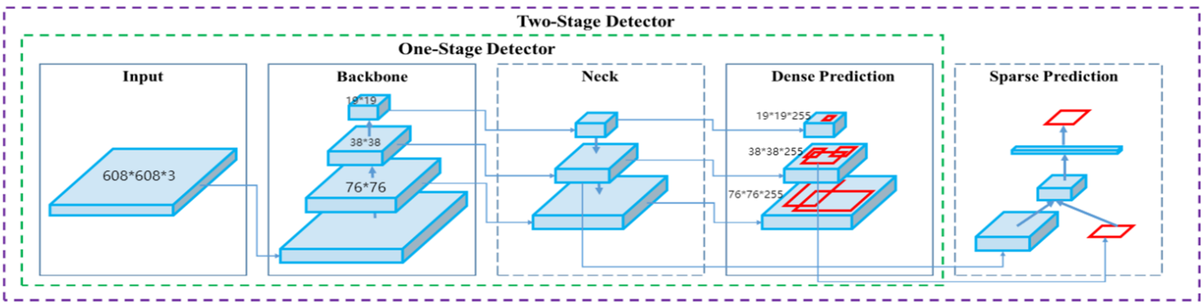


Fig 1. Object detector of YOLOv4 [12]

3. Method

Our innovation is that we use a pure computer vision solution to improve vehicle detection accuracy, which means that autonomous driving in extreme weather such as fog can be achieved without the need for expensive other equipment such as lidar. We do more information recovery by adding a dehazing module to the YOLO model. With modular preprocessing, the trained model can better improve the accuracy of vehicle detection. We use the Multi-Scale Retinex [13] with Color Restore or MSRCR [14] algorithm in the Retinex image enhancement algorithm to dehaze the foggy image of the training set to obtain an enhanced set of the training set.

The formula of the MSRCR algorithm is shown in the following.

$$R_{MSRCR_i}(x,y) = 255\frac{R_{MSRCR_i}(x,y) - min_i\left(min_{(x,y)}R_{MSRCR_i}(x,y)\right)}{max_i\left(max_{(x,y)}R_{MSRCR_i}(x,y) - min_i\left(min(x,y)R_{MSRCR_{(x,y)}}\right)\right)} \quad (1)$$

The following figure shows the main function of the python implementation method of the MSRCR algorithm.

The parameters used in this experiment are: {“sigma_list”: [15, 80, 200], “G”: 5.0, “b”: 25.0, “alpha”: 125.0, “beta”: 46.0, “low_clip”: 0.01, “high_clip”: 0.99}

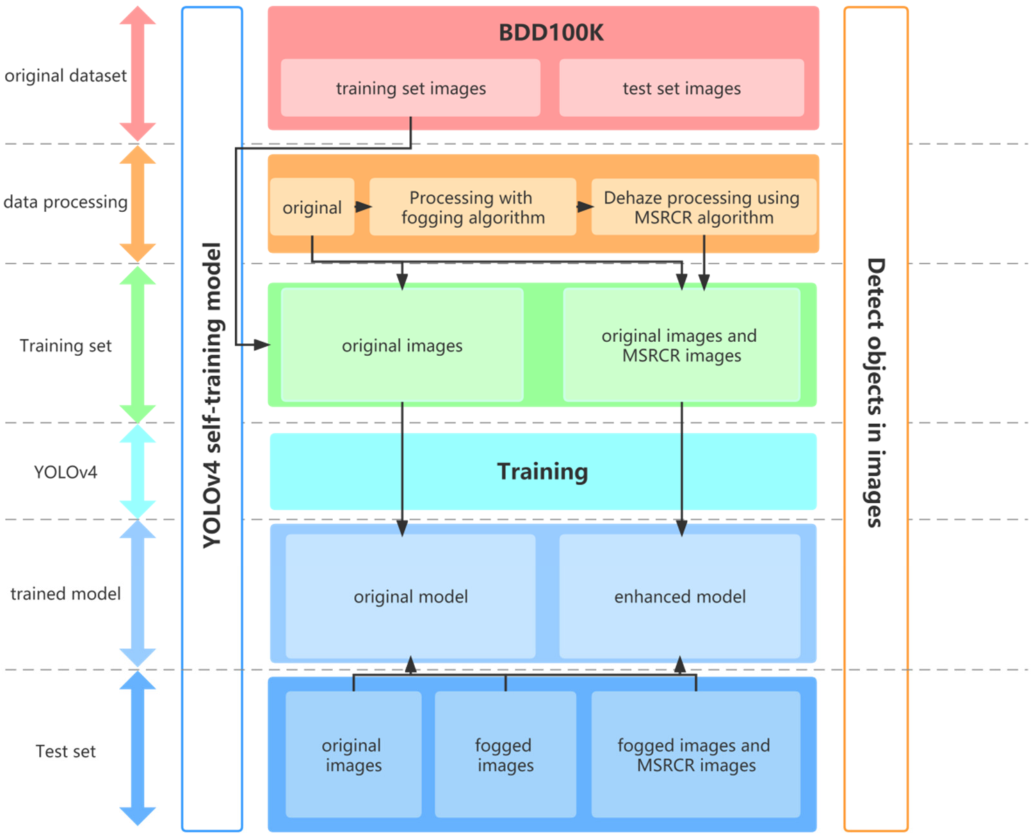


Fig 2. Training, testing, and overall experiment flow chart

Our original dataset comes from BDD100K[15]. The introduction and detailed implementation of the dataset will be explained in the next section. We split the original dataset into a training set and a test set. Two training sets are formed by preprocessing such as fogging and MSRCR algorithm dehazing: original images and original images plus MSRCR images. The next step is to train the model using YOLOv4. After setting up the data, names, and cfg files of the two groups respectively, we use the yolov4.conv.137 file to start training. So far, two kinds of self-training models have been obtained, namely the original model and the enhanced model. After we put the prepared three test sets into the two models obtained from training, we will get six sets of results. The comparison of these six groups of results can lead to the conclusion of this experiment.

4. Experiments

4.1. Experiment setup

This experiment was completed on ubuntu20.04 with a GeForce RTX™ 2080 Ti. The graphics card driver version of the experimental environment is 460, the cuda version is 11.3, the cudnn version is 8.2.1, and the opencv version is 3.4.16.

This paper uses Average Precision (AP) to evaluate the performance of the model.

AP is the area enclosed by the curve with the x-axis and y-axis, i.e. the integral of the PR curve. Calculated as follows:

$$AP = \int_0^1 p(r)dr \tag{2}$$

4.2. Dataset

BDD100K The dataset and annotation set used in this paper is from BDD100K. BDD100K is A Large-scale Diverse Driving Video Database from Berkeley Artificial Intelligence Research Lab. It has collected 100,000 driving videos from over 50,000 rides. Each video is 40 seconds long at 30 frames per second. Over 100 million frames in total. It samples the 10th-second keyframe of each video to get 100,000 pictures (picture size: 1280*720) and annotates them. It includes multiple scene types including city streets, residential areas, and highways, as well as different weather conditions at different times of the day.

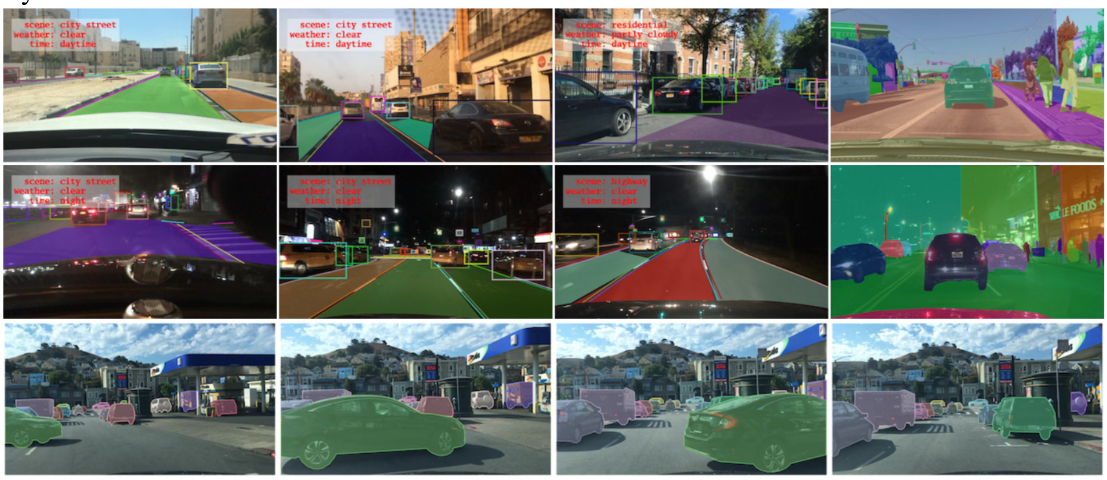


Fig 3. Overview of BDD100K dataset [15]

350 images in different scenes were selected from BDD100K as the original images of the training set.

We use these 350 photos to get the foggy training set by adding fog algorithm. Then we use the MSRCR algorithm to dehaze the resulting foggy training set to obtain an enhanced MSRCR training set.

So far there are two training sets:1. original image 2. original image and MSRCR image.

Among the remaining photos of BDD100K, 150 photos of different scenes are selected as the original images of the test set. On the basis of the original image, the fogging algorithm is used to obtain the foggy test set. I then used the MSRCR algorithm to get an enhanced test set based on the foggy test set. The resulting enhanced test set and foggy test are set together to form a third test set.

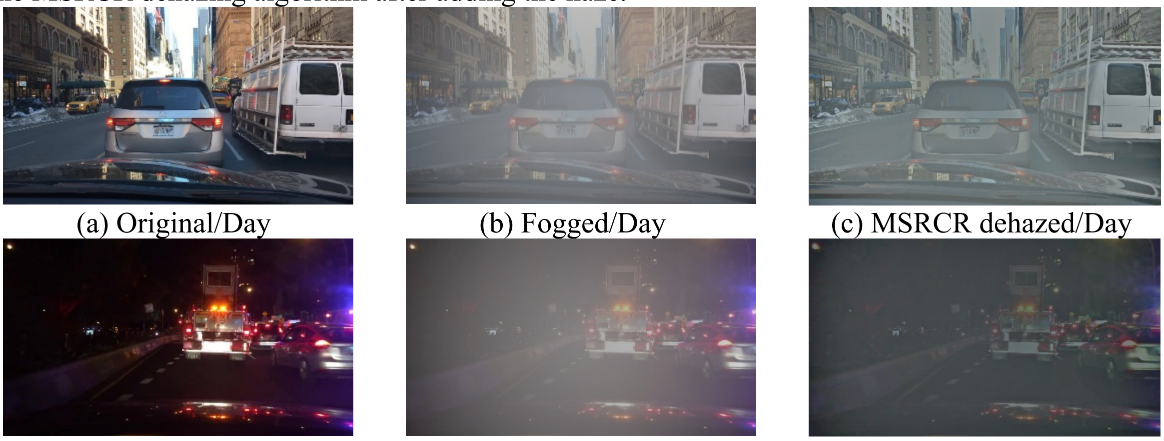
So far there are three test sets: 1. original image 2. fogged image 3. fogged image and MSRCR image.

Data annotation:

First, I convert the json annotation file provided by BDD100K into voc format xml. Then convert the xml file into the txt format required by YOLO. Finally, generate a txt file of the file path collection of each group of photos in the training set and test set. The resulting txt file will be filled in train and valid in voc.data.

4.3. Experiment evaluation

This experiment uses python to implement the fog adding algorithm and the MSRCR dehazing algorithm. This experiment uses python to implement the fog adding algorithm and the MSRCR dehazing algorithm. Shown below are the results of the original image and the original image using the haze algorithm and the MSRCR dehazing algorithm after adding the haze.



(a) Original/Day (b) Fogged/Day (c) MSRCR dehazed/Day

(d) Original/Night (e) Fogged/Night (f) MSRCR dehazed/Night

Fig 4. Original image, fogged image, MSRCR dehazed image during day and night

This paper compares six sets of experiments.

(a) Original (b) Fogged (c) Fogged and MSRCR

Fig 5. Example of detection results of the original model

(a) Original (b) Fogged (c) Fogged and MSRCR

Fig 6. Example of detection results of the enhanced mode

This paper uses the Average precision (AP) value to evaluate the performance of the model.

Table 1. The AP results from all six sets of experiments

No.	Training	Test	AP
1	original	original	0.8303
2	original	fogged	0.7181

3	original	fogged and MSRCR	0.7532
4	original and MSRCR	original	0.8253
5	original and MSRCR	fogged	0.8182
6	original and MSRCR	fogged and MSRCR	0.8212

The first group has the highest AP because the properties of the content of its training and test sets are very similar. The second group has the lowest AP because the properties of the content of the training set and test set are quite different. It can be seen from these two groups that the difference between the training set and the test set does have a great influence on the detected AP value. It can be seen from this set of experiments that the vehicle detection accuracy obtained by testing the fogged test set with the model trained on the original training set is low.

Comparing the second and third sets of experiments, it can be seen that the MSRCR algorithm used in the test set does work.

After adding the MSRCR training set, the AP values of groups 5 and 6 increased but group 4 decreased. Obviously, compared with the first group, the attribute gap between the training set and the test set of the fourth group has become larger.

The close accuracy of the first and fourth groups indicates that the preprocessed training set of the MSRCR algorithm is similar to the original image. This shows that the dehazing algorithm of the MSRCR algorithm is very effective and close to the original image.

Compared with the second group, the fogged test set is also tested, and the training set preprocessed by the MSRCR algorithm is added, and the accuracy is greatly improved. This shows that the MSRCR algorithm is very effective for improving the detection accuracy of vehicles in foggy weather.

The sixth group obtained higher test accuracy because the training and test sets were the same.

Comparing the three groups of four, five, and six, it can be seen that the MSRCR algorithm is indeed effective whether it is preprocessing in the training set or the test set, but there is still some gap compared with the original image.

5. Conclusion

Aiming at the insufficient accuracy of vehicle detection technology in foggy weather, this paper uses only computer vision to improve the accuracy of vehicle detection.

We used the BDD100K data source, involving different practical application scenarios, using the MSRCR algorithm for preprocessing, and using the YOLOv4 model for training, resulting in a vehicle detection model with improved accuracy. The results show that the model improves the accuracy in different degrees of foggy weather. This makes sense for the availability of driverless vehicles in foggy weather at less cost. In the future, this technology can be combined with low-cost radar for further accuracy improvements.

References

[1] Li, Q. L., & He, J. F. (2011). Vehicles detection based on three-frame-difference method and cross-entropy threshold method. Computer Engineering, 37(4), 172-174.

[2] Liu, Y., Lu, Y., Shi, Q., & Ding, J. (2013, December). Optical flow based urban road vehicle tracking. In 2013 ninth international conference on computational intelligence and security (pp. 391-395). IEEE.

[3] Radhakrishnan, M. (2013). Video object extraction by using background subtraction techniques for sports applications. Digital Image Processing, 5(9), 91-97.

[4] Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Competition and cooperation in neural nets (pp. 267-285). Springer, Berlin, Heidelberg.

[5] Cai, Z., Fan, Q., Feris, R. S., & Vasconcelos, N. (2016, October). A unified multi-scale deep convolutional neural network for fast object detection. In European conference on computer vision (pp. 354-370). Springer, Cham.

[6] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
[7] Kaiming, H., Xiangyu, Z., & Shaoqing, R. (2014). Jian Sun "Spatial pyramid pooling in deep convolutional networks for visual recognition‖ European". In Conference on Computer Vision.
[8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
[10] Singh, R., Singh, S., & Kaur, N. (2016). A Review: Techniques of vehicle detection in Fog. Indian Journal of Science and Technology, 9(45).
[11] Qian, K., Zhu, S., Zhang, X., & Li, L. E. (2021). Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 444-453).
[12] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[13] Land, E. H., & McCann, J. J. (1971). Lightness and retinex theory. Josa, 61(1), 1-11.
[14] Wang, J., Lu, K., Xue, J., He, N., & Shao, L. (2017). Single image dehazing based on the physical model and MSRCR algorithm. IEEE Transactions on Circuits and Systems for Video Technology, 28(9), 2190-2199.
[15] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., ... & Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2636-2645).