Vehicle target detection in complex scenes based on YOLOv3 algorithm

Lecheng Ouyang¹, Huali Wang¹*

¹ College of Communication Engineering, PLA Army Engineering University, Nanjing, Jiangsu, 210007, China
*Corresponding author’s e-mail: huali.wang@ieee.org

Abstract. In view of the low accuracy of traditional vehicle target detection methods in complex scenes, combined with the current hot development of deep learning, this paper applies the YOLOv3 algorithm framework to achieve vehicle target detection. By using PASCAL VOC2007 and VOC2012 data sets, images containing vehicle targets were screened out to constitute the VOC car data set, and the target detection problem was converted into a binary classification problem. Then loading the pre-trained YOLOv3 model weight, and training the vehicle target detection model weight based on YOLOv3 algorithm, which is used to detect the test samples. Experimental results show that this method has advantages over the traditional target detection algorithms in recognition accuracy and detection speed.

1. Introduction

In recent years, with the popularity of family cars, traffic congestion is increasing, and the demand for intelligent traffic monitoring system is increasing, among which the real-time detection technology of vehicle targets is particularly critical. With the development of unmanned driving technology, real-time detection of vehicle targets is also the primary problem to be solved. In the actual scene, the vehicle target in the picture or video frame will encounter complex problems such as occlusion, background noise, light mutation, shooting Angle, etc., which brings great difficulties to the traditional target detection methods, which will bring bad effect on detection. Therefore, different from traditional detection methods and combined with the current rapidly developing in deep learning detection algorithm, this paper proposes a vehicle target detection method based on YOLOv3 algorithm in complex scenes.

Traditional vehicle target detection methods include: frame difference method[1], optical flow method[2], background difference method[3].These methods are mainly based on the change of information between video frames, rather than detection and recognition through the characteristics of vehicles. Such methods are fast, but have low detection accuracy and poor robustness for complex scenes. Therefore, machine learning method extracts target features, such as HOG[4] features, SIFT[5] and other methods, then inputs the extracted features into classifiers, such as SVM[6], for classification detection. Those method face two problems: one is that the features are set artificially, and appropriate features need to be extracted, the other is that the design of classifier also determines the final target detection effect, which is the bottleneck of the current machine learning method for target detection.

With the rapidly development of deep learning theory and practice, target detection methods based on deep learning algorithm emerge rapidly. Different from the traditional feature extraction and matching algorithm, the deep convolution network has a certain degree of tolerance for geometric
transformation, deformation and illumination. Driven by the training data, it can construct the feature description adaptively and has a good flexibility and expansibility. In 2013, R-CNN was born. As a pioneer in the application of deep learning method in the field of target detection, the mAP value of VOC2007 was improved to 48%. In 2014, the mAP value was increased to 66% by modifying the network structure, and the mAP value of ILSVRC2013 test set was also improved to 31.4%. Girshick[7-8] proposed the R-CNN, which has made a breakthrough in the field of target detection. SPP-net [9], Fast R-CNN[10], Faster R-CNN[11], R-FCN[12], YOLO[13], SSD[14] and other algorithms have been developed, which have greatly improved the accuracy and speed of target detection. Training on Titan X, at the same mAP value, YOLOv3 [15] is 3.8 times faster than RetinaNet [16]. What’s more, YOLOv3 can carry out a picture of 320×320 in 22 millisecond, and the mAP value is 51.5. With the same mAP value of the SSD, YOLOv3 is three times faster. Therefore, according to the characteristics of vehicle targets in complex scenes, this paper produces VOC vehicle data set and applies the YOLOv3 algorithm to vehicle target detection in complex scenes.

2. Vehicle target detection

The vehicle target detection process designed in this paper is shown in figure 1. Firstly, screening out samples which contain car targets from the data sets of VOC2007 and VOC2012 to constitute the VOC-car data sets. Then putting the training samples into the YOLOv3 training network, and the training is carrying out until the network convergence. Finally, the weight.h5 file was loaded into the YOLOv3 model to test samples.

![Flow chart of car targets detection](image.png)

Figure 1. Flow chart of car targets detection.

2.1. YOLOv3 introduction

Proposed by Redmon et al. in 2016, the YOLO algorithm is a convolutional neural network that can predict multiple Box locations and categories at one time. In a real sense, it realizes the end-to-end target detection and plays the advantage of fast speed, but its accuracy decreases. However, the YOLO9000[17] algorithm is an improvement on the original YOLO algorithm, which maintains the speed and improves the accuracy at the same time. In this way, YOLO9000 can simultaneously train in the data sets of COCO and ImageNet, and the trained model can detect 9000 objects.

YOLOv3 predicts four coordinate values for each of the bounding box, which is based on the top left corner of the image offset, as well as the bounding box by wide and high. YOLOv3 predicts the score of an object for each bounding box by logistic regression. If the prediction of the bounding box coincide better with real border than others, then its value is one. Compared with the previous YOLO algorithm, YOLOv3 has made many improvements. Firstly, it carries out multi-scale prediction, and predicts three boxes for each scale. The design method of anchor still uses clustering method, and obtains nine cluster centers, which are divided into three scales according to size. Secondly, YOLOv3 choose better basic classification network and classifier darknet-53[15].Thirdly, Softmax is replaced by logistic regression for classifier in category prediction. Figure 2 is the structure diagram of YOLOv3 detection.
| Type    | Filters | Size   | Output       |
|---------|---------|--------|--------------|
| Convolutional | 32      | 3×3    | 256×256      |
| Convolutional | 64      | 3×3/2  | 128×128      |
| Residual  | 32      | 1×1    | 128×128      |
| Convolutional | 64      | 3×3    | 128×128      |
| Residual  | 256     | 3×3/2  | 64×64        |
| Convolutional | 128     | 1×1    | 32×32        |
| Convolutional | 256     | 3×3    | 32×32        |
| Residual  | 512     | 3×3/2  | 16×16        |
| Convolutional | 256     | 1×1    | 16×16        |
| Residual  | 512     | 3×3    | 16×16        |
| Convolutional | 1024    | 3×3/2  | 8×8          |
| Residual  | 1024    | 1×1    | 8×8          |
| Avgpool  | 1000    |        |              |

Figure 2. YOLOv3 detection structure diagram.

2.2. Making VOC-car data set

The PASCAL VOC dataset provides about 10,000 images for image recognition and classification, which are four categories of people, animals, vehicles, and indoor items. The three folders used for target detection in the PASCAL VOC dataset are mainly Annotations, ImageSets, and JPEGImages. In the JPEGImages folder, all the training set images are stored. The file format is jpg. In Annotations folder, the annotation information of all training images are stored. The content of the xml file contains the name, width, height, number of channels, and the name of the target in the image, the position of the target in the image, etc. The ImageSets folder stores train.txt, test.txt, trainval.txt and val.txt, which store the training, testing, training verification, and verification sample names.

Data training is the key in deep learning networks. In this paper, for the vehicle target detection, the pictures containing the vehicle targets are selected from the PASCAL VOC2007 and VOC2012 data sets to make a separate VOC-car data set. Write a script file to determine whether there is a vehicle target in the image based on whether the content of the xml file contains <name>car</name>, so as to filter out the image containing the vehicle target. However, the image may contain other targets besides the car target, such as <name>person</name>. Therefore, it is necessary to clear the relevant line of <name> not in the car, and finally complete the production of the VOC-car data set.

The VOC-car dataset produced in this paper includes: 2765 jpg images and 2765 xml files. In the VOC-car dataset, there are various vehicle targets with different perspectives, and different perspectives are selected, whose number are shown in Table 1. The vehicle image samples from different perspectives are shown in Figure 3.
Table 1. Number of data sets from different perspectives.

| Perspective    | Training set | Test set | Training validation set | Validation set |
|----------------|--------------|----------|--------------------------|----------------|
| Front view     | 578          | 168      | 251                      | 80             |
| Behind view    | 287          | 74       | 101                      | 29             |
| Side view      | 983          | 310      | 437                      | 128            |
| Look down      | 64           | 21       | 29                       | 8              |
| Internal view  | 24           | 7        | 11                       | 4              |

Figure 3. Vehicle image samples from different perspectives.

2.3. Training model
After the VOC-car dataset is completed, the YOLOv3 model is trained. The training host is configured as: Intel i9-9900k CPU@3.60GHz processor, 64G memory, and NVIDIA RTX2080Ti discrete graphics card. For the vehicle target detection task in complex scenes, this paper re-trains two kinds of real-time deep learning models YOLOv3-tiny and YOLOv3. YOLOv3-tiny is a simplified version of YOLOv3. The former has less convolution layer in the network structure than the latter. Although the former has reduced a small part of the accuracy, but the speed is much improved, it is very convenient to embed in the real-time detection platform.

Before training, we need to configure the relevant files and network structure. First set the classes in the voc.annotation file to 1, indicating that only one type of target is identified. Then the classes in the yolov3.cfg configuration file are changed to 1, and the values in the last layer of the convolutional layer are changed to \(3 \times (\text{number of categories} + 5) = 3 \times (1+5) = 18\), which it to say that, each box contains 3 boxes for detection, each box outputs identification category, 4 coordinate information, and confidence. Finally, change the recognition type in the coco_classes.txt and voc_classes.txt files to car.

Start training. We can choose to start training from random initialization of weights, but such training will take some time and not necessarily get the desired result. So you can use the migration learning method to load the pre-trained on the ImageNet dataset.

3. Experimental results and analysis
The YOLOv3 algorithm uses the typical Darknet-53 model to train and fine-tune the parameters of the network. In the experiment, the pre-training model on the ImageNet dataset is used to initialize the Darknet-53 model. With the iterative training, the network reaches convergence. After the training completed, test the sample picture. In figure 4 and figure 5, the left side is the original picture, and the
right side is the test picture. It can be seen from the figure that for the complex environment vehicle target detection, the detection algorithm based on YOLOv3 is more accurate than the detection algorithm based on YOLOv3-tiny. The former can better detect the target position, while the latter is prone to false detection in complex environments such as occlusion and poor light.

Figure 4. Vehicle target detection effect diagram based on YOLOv3 algorithm.

Figure 5. Vehicle target detection effect diagram based on YOLOv3-tiny algorithm.

During the test, the IOU values of the bounding box and the reference standard box are detected to judge the real example, the false positive example and the false counterexample in the result. When the IOU ≥ 0.5, it is a real example. When the IOU < 0.5, it is false. When the IOU = 0, it is a false counterexample.

The experimental results are shown in Table 1. The two detection algorithms are applied to vehicle target detection in complex scenes, which have advantages in speed and accuracy. The AP value based on YOLOv3 reaches 89.16%, which fully demonstrates the precision advantage of the YOLOv3 algorithm for target detection, but the detection method based on the YOLOv3-tiny algorithm is faster and the accuracy is only slightly inferior. If applied to real-time target detection, the detection method based on YOLOv3-tiny algorithm is more suitable. If the target is detected by the background computer, the detection method based on the YOLOv3 algorithm is better.

Table 2. Comparison of two vehicle target detection methods.

| Vehicle target detection methods          | AP(%) | Recall(%) | FPS |
|------------------------------------------|-------|-----------|-----|
| Based on YOLOv3 algorithm                | 89.16 | 84.59     | 21  |
| Based on YOLOv3-tiny algorithm           | 76.82 | 78.84     | 33  |
4. Conclusion
In this paper, the deep learning YOLOv3 target detection algorithm is applied to vehicle target detection in complex scenes. Through the analysis and comparison of experimental results, the accuracy of this method can reach 89.16%, and the running speed averages 21FPS. Compared with traditional target detection, the method is greatly improved in accuracy or operational efficiency. In addition, the detection methods of this paper proposed can also be applicable to other target detection, but the premise is that a large amount of data is needed for model training.

References
[1] Seki, M., Fujiwara, H., Sumi, K. (2002) A robust background subtraction method for changing background. In: IEEE Workshop on Applications of Computer Vision. Orlando. pp. 207-213.
[2] Barron, J.L., Fleet, D.J., Beauchemin, S.S. (1994) Performance of optical flow techniques. International Journal of Computer Vision., 12(1): 43-77.
[3] Horprasert, T., Harwood, D., Davis L.S. (1999) A statistical approach for real-time robust background subtraction and shadow detection. In: IEEE International Conference on Computer Vision. Kerkrya. pp. 99: 1-19.
[4] Taigman, Y., Yang, M., Ranzato, M.A., Wlof, L. (2014) Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus. pp. 1701-1708.
[5] Ma, X., Grimson, W. E. L. (2005) Edge-based rich representation for vehicle classification. In: Tenth IEEE International Conference on Computer Vision. Beijing. pp. 2: 1185-1192.
[6] Kazemi, F.M., Samadi, S., Poorreza, H.R., Akbarzadeh-T, M.R. (2007) Vehicle recognition using curvelet transform and SVM. In: Fourth International Conference on Information Technology. Rourkela. pp. 516-521.
[7] 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. Columbus. pp. 580-587.
[8] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2016) Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(1): 142-158.
[9] He, K., Zhang, X., Ren, S., Sun, J. (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9): 1904-1916.
[10] Girshick, R. (2015) Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. Santiago. pp. 1440-1448.
[11] Ren, S., He, K., Girshick, R., Sun, J. (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. Montréal. pp. 91-99.
[12] Dai, J., Li, Y., He, K., Sun, J. (2016) R-FCN: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems. Barcelona. pp. 379-387.
[13] 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. Las Vegas. pp. 779-788.
[14] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016) SSD: Single shot multibox detector. In: European Conference on Computer Vision. Amsterdam. pp. 21-37.
[15] Redmon, J., Farhadi, A. (2018) Yolov3: An incremental improvement. https://arxiv.org/abs/1804.02767.
[16] Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. Venice. pp. 2980-2988.

[17] Redmon, J., Farhadi, A. (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Hawai. pp. 7263-7271.