Vehicle detection in aerial image based on deep learning

Shuai Zhao¹, Fucheng You¹, Lei Shang¹, Chao Han¹ and Xuewei Wang¹

Beijing Institute of Graphic Communication. No. 1, Xinghua Street (two sections), Daxing District, Beijing
Email: 17839192463@163.com

Abstract. At present, urban traffic congestion is becoming more and more serious. Traditional vehicle detection methods generally have problems of low precision and low recognition rate. In order to solve these problems, a vehicle detection method in aerial image based on deep learning is proposed. The method uses the YOLOv3 algorithm and optimizes the algorithm network, increase the number of rasters of the final predicted output, improves the detection ability of the network for dense groups, and verifies the effect by training a new model. The experimental results show that the new training model has a good effect on vehicle detection, which solves the problem of low detection and recognition rate of urban traffic vehicle detection, and makes good predictions and judgments on urban traffic congestion.

1. Introduction

As urban traffic congestion becomes more and more serious, people's demands for real-time detection of road traffic information also increase. Through vehicle detection and analysis of road traffic and congestion information, road conditions can be provided to drivers in a timely manner[1-2]. In recent years, many algorithms and techniques have been proposed at home and abroad. For example, Hedi Harzallah[3] and others proposed a vehicle detection method using a sliding window. Usually, in order to prevent missing detection, the sliding window will adopt smaller window interval and window scale multiple when traversing the image, but this will lead to a decrease in the real-time performance of the method.

In addition, with the proposed target detection algorithm based on depth learning, R-CNN[4] detection is more effective. This method combines selective search, CNN, SVM and bounding box regression to significantly improve the detection performance. The vehicle detection algorithm based on depth learning combines the Faster R-CNN[5] open source framework and LOCNET network algorithm. The use of deep learning convolution neural network can make up for the defect that the applicability of manually designed vehicle target features is not extensive and improve the accuracy of vehicle target detection and location[6].

In 2015, Redmon J proposed YOLO [7] detection algorithm. YOLO is a new End-to-End detection algorithm. Although YOLO also belongs to CNN, YOLO blurs out the differences among CG, FE and CV3 stages in the detection process and directly and quickly completes the detection task. YOLO can guarantee both accuracy and detection rate. Standard YOLO can detect 45 pictures per second, while Fast YOLO can detect 155 f/s at the same time.
2. Introduction to Algorithm

2.1 Algorithm Principle
The basic idea of YOLO [7] algorithm is to first extract features from an input image through a feature extraction network to obtain a feature map of a certain size, such as 13*13, then divide the input image into 13*13 grid cells, and then if the center coordinates of an object in ground truth fall in which grid cell. Then the object is predicted by the grid cell, because each grid cell predicts a fixed number of bounding boxes (2 in YOLOv1 and v2 and 3 in YOLOv3, the initial size of these bounding boxes are different from each other), and only the bounding box with the largest IOU of ground truth can be used to predict the object. It can be seen that the predicted output feature map has two dimensions, such as 13*13, and another dimension (depth) is B*(5+C).

2.2. Advantages of the Algorithm
YOLO algorithm, especially YOLOv3 [8] algorithm, has very good accuracy and detection rate. YOLOv3 algorithm runs faster than other detection methods under the same equipment and data sets. Figure 1 below is a comparison of YOLOv3 with other algorithms:

![Algorithm comparison result](image)

Figure 1. Algorithm comparison result

3. Algorithm Improvement

3.1 Bounding Box Prediction
Bounding boxes coordinate prediction method continues to use YOLOv2 approach[9], tx, ty, tw and th are the predicted outputs of the model. cx and cy represent the coordinates of grid cell. for example, if the feature map size of a layer is 13*13, then grid cell has the size of 13*13, the coordinates cx of grid cell in row 0, and column 1 are 0 and cy is 1. pw and ph represent the size of the bounding box before prediction. bx, by, bw and bh are the predicted coordinates and size of the center of the bounding box. As shown Figure 2 and Figure 3:
YOLOv3 uses logical regression to predict the target score for each bounding box. If the previous bounding box overlaps the object more than any other bounding box before, the value should be 1. If the bounding box is not the best first, but overlaps with the object by more than a certain threshold, the prediction is ignored, followed by [5], and the threshold of 5 is used. Unlike [5], the algorithm specifies only one bounding box for each object.

3.2 Class Prediction

The improvement in category prediction lies in the improvement of the original single label classification to multi-label classification, so the original softmax layer used for single label multi-classification is replaced by the logical regression layer used for multi-label multi-classification. First of all, explain the reason why we want to make such a modification. The original softmax layer in the classification network assumes that an image or an object belongs to only one category, but in some complex scenarios, an object may belong to multiple categories, such as woman and person in your category. If there is a woman in an image, then the category labels in your test results will have woman and person categories at the same time. This is multi-label classification, and it is necessary to use the logistic regression layer to make two classifications for each category. The logistic regression layer mainly uses sigmoid function, which can constrain the input to the range of 0 to 1.

3.3 Predictions Across Scales

YOLOv3 is predicted by using multiple scale fusion. In the YOLOv2 algorithm, there is a network layer called pass through layer. Assuming the size of the last extracted feature map is 13*13, the role of this layer is to connect the 26*26 feature map of the previous layer with the 13*13 feature map of this layer, a bit like ResNet[10]. This was also done to enhance the accuracy of the YOLO algorithm in detecting small targets. This idea has been further strengthened in YOLOv3, where the upsample and fusion method similar to FPN are adopted (finally, three scales are fused, and the other two scales are 26*26 and 52*52, respectively), and the detection effect of small targets is improved obviously when tested on feature map of multiple scales.

3.4 Feature Extractor

YOLOv3 uses a new network for feature extraction. The new network is a hybrid method between the networks used in YOLOv2 and Darknet-19 and those novel remaining networks. The network uses the successful 3*3 and 1* translation layers, but now there are some quick connections that are very large. Each network is trained with the same set and test data 256*256 to achieve single crop accuracy. Therefore, Darknet-53 runs on the most advanced classifier, but with fewer floating point operations and more speed. Darknet-53 is better than ResNet-101 and 1.5* faster. Darknet-53 has similar performance to ResNet-152 and is 2* fast. The network diagram of YOLOv3 is shown in Figure 4:
3.5. Network Improvement

In this paper, the network architecture has been further modified. By increasing the number of grids finally predicted, the network's ability to detect fine-grained features and distinguish different objects has been improved, and the detection effect of dense object groups has been improved. Algorithm and network architecture diagram as shown in Figure 5:

Figure 4. Network composition

| Type       | Filters | Size       | Output       |
|------------|---------|------------|--------------|
| Convolutional | 32      | 3 x 3      | 256 x 256    |
| Convolutional | 64      | 3 x 3 / 2  | 128 x 128    |
| Convolutional | 32      | 1 x 1      |              |
| Convolutional | 64      | 3 x 3      |              |
| Residual    |         |            | 128 x 128    |
| Convolutional | 128     | 3 x 3 / 2  | 64 x 64      |
| Convolutional | 64      | 1 x 1      |              |
| Convolutional | 128     | 3 x 3      |              |
| Residual    |         |            | 64 x 64      |
| Convolutional | 256     | 3 x 3 / 2  | 32 x 32      |
| Convolutional | 128     | 1 x 1      |              |
| Convolutional | 256     | 3 x 3      |              |
| Residual    |         |            | 32 x 32      |
| Convolutional | 512     | 3 x 3 / 2  | 16 x 16      |
| Convolutional | 256     | 1 x 1      |              |
| Convolutional | 512     | 3 x 3      |              |
| Residual    |         |            | 16 x 16      |
| Convolutional | 1024    | 3 x 3 / 2  | 8 x 8        |
| Convolutional | 512     | 1 x 1      |              |
| Convolutional | 1024    | 3 x 3      |              |
| Residual    |         |            | 8 x 8        |
| Avgpool     |         | Global     |              |
| Connected   |         | 1000       |              |
| Softmax     |         |            |              |

Figure 5. Algorithm and network architecture diagram

"It struggles to generalize objects in new or unusual aspect ratios or configurations"

"Our model struggles with small objects that appear in groups, such as flocks of birds"

"Our model uses relatively coarse features for predicting bounding boxes since our architecture has multiple downsampling layers"

Maximum input size of “600 pixels"

Augment training data with image rescaling and rotations

* New network architecture with finer grained features and a denser final grid

* Up-sample to look for small, densely packed objects

Run an ensemble of detectors at multiple scales

Intelligently partition and recombine large test images
4. Experiments and Results Analysis

4.1 Experimental data
The data used in the experiment are mainly from aerial images, with a resolution of about 1000 * 600, and a total of 2,000 images, of which 1,500 images are used as training sets and the remaining 500 images are used as data sets. The training set and data set are marked manually in VOC data set format, and vehicles are marked one by one by using labeling.

4.2 Experimental platform
The hardware configuration processor is GPU (GTX-1080) with 32G. The software development environment is Windows10 + CUDA 9.1 + CUDNN7, and the applied language is mainly C++.

4.3 Comparison of experimental results
It is mainly divided into comparison of detection effects between different algorithms and comparison of detection effects between different data sets, as shown in Figure 6 and Figure 7:

Figure 6. different data sets comparison results
5. Conclusion
The YOLOv3 algorithm is applied to realize vehicle detection in aerial images, which solves the problems of insufficient real-time detection of vehicles in the past and improves YOLOv3 network framework, thus improving the detection capability for dense groups. However, there are still some problems in vehicle detection, such as low recognition rate and difficult recognition in certain specific environments. New measures will be taken to improve YOLO algorithm network in the next step to improve the robustness of the network and enhance its recognition effect in complex environments.

6. Acknowledgment
This work was partially supported by Joint Funding Project of Beijing Municipal Commission of Education and Beijing Natural Science Fund Committee (KZ201710015010), Project of National Scientific Found (No.61370188), Project of Beijing Municipal College Improvement Plan (PXM2017_014223_000063) and New Project of Green Printing and Publishing Technology by Cooperative Creating Center (PXM_014223_000025) and BIGC Project (Ec201803 Ed201802 Ea201806).

7. References
[1] WANG Yuning, PANG Zhiheng, YUAN Deming. Vehicle Detection Based on YOLO in Real Time [J]. Journal of Wuhan University of Technology, 2016, 38(10):41-46.
[2] Bautista C M.Dy C A, Mañalac M I, et al. Convolutional Neural Network for Vehicle Detection in Low Resolution Traffic Videos[C]//Region 10 Symposium(TENSYMP).2016 IEEE.[S.L.]: IEEE, 2016:277-281.
[3] Harzallah H, Schmid C, Jurie F, et al. Classification Aided Two Stage Localization[C]// PASCAL Visual Object Classes Challenge Workshop, in Conjunction with ECCV.2008.
[4] Fan Q, Brown L, Smith J. A Closer Look at Faster R-CNN for Vehicle Detection[C]// Intelligent Vehicles Symposium(IV).2016 IEEE.[S.I.]:IEEE.2016:124-129.
[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal net-works. arXiv preprint arXiv:1506.01497, 2015. 2
[6] GIRSHICK R. Fast R-CNN[C]//IEEE International Conference on Computer Vision. Santiago:IEEE, 2015:10-15

[7] J. Redmond, 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.[S.l.]:IEEE, 2016.779788.

[8] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018. 4

[9] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017. 1, 2, 3

[10] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 2, 3