Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 98-107
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Vehicle Detection Method for Aerial Image Based on YOLO

Junyan Lu[1*], Chi Ma[2], Li Li[2], Xiaoyan Xing[2], Yong Zhang[2], Zhigang Wang[2], Jiuwei Xu[2]

[1]Chang Guang Satellite Technology Co., Ltd., Changchun, China
[2]Department of Transportation of Jilin Province, Changchun, China
Email: *lujy1990@sina.com

How to cite this paper: Lu, J.Y., Ma, C., Li, L., Xing, X.Y., Zhang, Y., Wang, Z.G. and Xu, J.W. (2018) A Vehicle Detection Method for Aerial Image Based on YOLO. *Journal of Computer and Communications*, **6**, 98-107.
https://doi.org/10.4236/jcc.2018.611009

Received: August 29, 2018
Accepted: November 12, 2018
Published: November 19, 2018

Abstract

With the application of UAVs in intelligent transportation systems, vehicle detection for aerial images has become a key engineering technology and has academic research significance. In this paper, a vehicle detection method for aerial image based on YOLO deep learning algorithm is presented. The method integrates an aerial image dataset suitable for YOLO training by processing three public aerial image datasets. Experiments show that the training model has a good performance on unknown aerial images, especially for small objects, rotating objects, as well as compact and dense objects, while meeting the real-time requirements.

Keywords

Vehicle Detection, Aerial Image, YOLO, VEDAI, COWC, DOTA

1. Introduction

In recent years, with the rapid development of information technology, intelligent transportation systems have become an important way of modern traffic management and an inevitable trend. As the key technology of intelligent transportation system, vehicle detection is the basis for realizing many important functions [1], such as measurement and statistics of traffic parameters such as traffic flow and density, vehicle location and tracking, and traffic data mining, etc.

At the same time, with the technology maturity and market popularization of UAV (Unmanned Aerial Vehicle), which has characteristics of being lightweight, flexible, and cheap, the aerial photography of UAVs in the application of scenes such as traffic information collection and traffic emergency response reflects a huge advantage.

In summary, vehicle detection for aerial image plays an important role in engineering applications. In addition, the technology relies on machine vision, artificial intelligence, image processing and other disciplines, and is a typical application of interdisciplinary research. Therefore, it also has important research significance in academics.

Based on YOLO deep learning algorithm and three public aerial image datasets, this paper presents a vehicle detection method for aerial image.

2. Related Work

The commonly used vehicle detection methods proposed by domestic and foreign scholars are mainly divided into three categories: based on motion information, based on features, and based on template matching. Cheng and others use background subtraction and registration methods to detect dynamic vehicles [2], Azevedo and others based on median background difference method to detect vehicles in aerial images [3]. The above two methods achieve the detection of moving objects, however, because the aerial video has the characteristics of complex scenes and diverse objects, the two methods cannot achieve the desired effect for accurate vehicle detection, and false and missed detection are also serious. Sivaraman and others combined Haar features and Adaboost to detect vehicles and implement vehicle detection on highways [4], Tehrani and others proposed a vehicle detection method based on HOG features and SVM to achieve vehicle detection in urban roads [5]. The above two methods improve the accuracy of detection, but since the traditional machine learning method only supports training for a small amount of data, there is still a shortage of detection of vehicle diversity.

In recent years, with the updating of computer hardware, especially GPU technology, the deep learning algorithms have been rapidly developed when solving problems in the fields of pattern recognition and image processing, and are more efficient and precise than traditional algorithms. Therefore, this paper uses a deep learning algorithm, YOLO, to achieve vehicle detection.

3. YOLO Deep Learning Object Detection Algorithm

YOLO, which has been proposed by Joseph Redmon and others in 2015 [6], is a real-time object detection system based on CNN (Convolutional Neural Network). On the CVPR (Conference on Computer Vision and Pattern Recognition) in 2017, Joseph Redmon and Ali Farhadi released YOLO v2 which has improved the algorithm's accuracy and speed [7]. In April this year, Joseph Redmon and Ali Farhadi proposed the latest YOLO v3, which has further improved the performance on object detection [8]. This chapter introduces the basic principles of the YOLO algorithm according to its update process.

3.1. YOLO v1

1) Basic idea

YOLO divides the input image into $S \times S$ grids. If the center coordinate of the GT (Ground Truth) of an object falls into a grid, the grid is responsible for detecting the object. The innovation of YOLO is that it reforms the Region Proposal detection framework: RCNN series need to generate Region Proposal in which to complete classification and regression. But there is overlap between Region Proposal, which will bring a lot of repetition work. However, YOLO predicts the bbox (bounding box) of the object contained in all grids, the location reliability, as well as the probability vectors of all classes at one time, thus it solves problem one-shot.

2) Network structure

YOLO network borrows Google Net while the difference is that YOLO uses the 1×1 convolutional layer (for cross-channel information integration) + 3×3 convolutional layer instead of the Inception module simply. YOLO v1 network structure consists of 24 convolution layers and 2 full connection layers, as shown in **Figure 1**.

3.2. YOLO v2

Compared with the region proposal based method such as Fast R-CNN, YOLO v1 has a larger positioning error and a lower recall rate. Therefore, the main improvements of YOLO v2 are to enhance the recall rate and positioning ability, and include:

1) BN (Batch Normalization)

BN is a popular training technique since 2015. By adding BN layer after each layer, the entire batch data can be normalized to a space with a mean of 0 and variance of 1, which can prevent the gradient from disappearing as well as gradient explosion, and make network convergence faster.

2) Anchor boxes

In YOLO v1, the full connection layer is used to predict the coordinates of bbox directly after the convolutional layer. YOLO v2 removes the full connection layer by using the idea of Faster R-CNN, and adds Anchor Boxes, which effectively improves the recall rate.

3) Multi-scale training

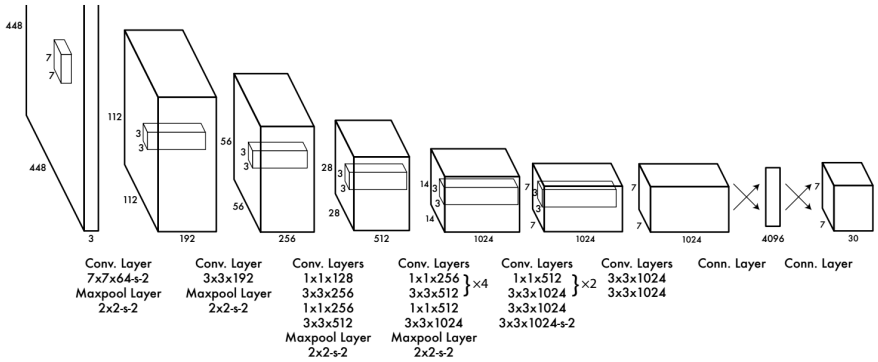

Figure 1. YOLO v1network structure.

The input image size for the YOLO v1 training network is fixed, where YOLO v2 adjusts the input image size randomly every 10 epoch during training, so that the model has a good detection effect on the multi-scale input images during the test.

3.3. YOLO v3

YOLO v3 model is much more complex than YOLO v2, and its detection on small objects, as well as compact dense or highly overlapping objects is very excellent. The main improvements include:

1) Loss

YOLO v3 replaces the Softmax Loss of YOLO v2 with Logistic Loss. When the predicted objects classes are complex, especially when there are many overlapping labels in the dataset, it is more efficient to use Logistic Regression.

2) Anchor

YOLO V3 uses nine anchors instead of the five anchors of YOLO v2, which improves the IoU.

3) Detection

YOLO v2 only uses one detection while YOLO v3 uses three, which greatly improves the detection effect on small objects.

4) Backbone

YOLO v3 replaces darknet-19 network of YOLO v2 with darknet-53 network, which improves the accuracy of object detection by deepening the network.

This paper uses the latest YOLO v3 model to achieve the vehicle detection for aerial image.

4. Public Datasets for YOLO Training

The performance of the classifier trained based on conventional dataset is poor on aerial images, because that aerial images have the following special features:

1) Scale diversity

The shooting height of UAVs ranges from tens of meters to kilometers, resulting in a wide range of size of similar object on the ground.

2) Perspective specificity

The perspectives of aerial images are basically high-altitude overlooking, while most of the conventional datasets are ground-level perspectives.

3) Small object

The objects of aerial images are generally only a few dozen or even a few pixels, so their amount of information is less also.

4) Multidirectional

Aerial images are taken from a bird's view, and the direction of objects are uncertain (while the object direction on the conventional dataset tends to have certainty, such as pedestrians are generally upright).

5) High background complexity

Aerial images have a large field of view (usually with a few square kilometers

of coverage), and it may contain a variety of backgrounds, which will have a strong interference with object detection.

For the above reasons, it is often difficult to train an ideal classifier on conventional datasets for the object detection tasks on aerial images. Therefore, a specialized aerial image dataset is needed. In this paper, three public aerial image datasets are used and processed to make a new aerial image dataset suitable for YOLO training. This chapter introduces the specific information of the three datasets.

4.1. VEDAI Dataset

The VEDAI (Vehicle Detection in Aerial Imagery) dataset is made by Sebastien Razakarivony and Frederic Jurie of University of Caen [9], whose original material is from the public Utah AGRC database. The raw images have 4 uncompressed color channels (three visible color channels and one near infrared channel). The authors firstly split the original large-field satellite image into 1024 × 1024 pixels JPEG format images, and then create the visible color channels dataset and the near infrared channel dataset, and finally down sample the above two datasets into 512 × 512 pixels, so VEDAI contains 4 subsets. In this paper, only the first subset of VEDAI (1024 × 1024, RGB 3 channels) is used. The shooting heights of all images in VEDAI are the same, and the GSD (Ground Sampling Distance) of 1024 × 1024 image is 12.5 cm pp (cm per pixel). VEDAI contains a total of 1250 images, and is manually annotated nine classes of vehicle ("plane", "boat", "camping car", "car", "pick-up", "tractor", "truck", "van", and "other"), a total of 2950 samples. The annotation of each sample includes: sample class, GT's center point coordinates, direction, and the coordinates of GT's 4 corners.

4.2. COWC Dataset

COWC (Cars Overhead with Context) dataset is made by T. Nathan Mundhenk and others of Lawrence Livermore National Laboratory [10], whose original materials are from six public websites. The COWC contains a total of 53 pictures in TIFF format, and the image size is between 2000 × 2000 to 19,000 × 19,000 pixels. COWC images have covered six geographic locations, namely Toronto (Canada), Selwyn (New Zealand), Potsdam and Vaihingen (Germany), Columbus and Utah (United States), in which the images of Vaihingen and Columbus are grayscale, while the others are in RGB color. The GSD of the image is 15 cmpp, so the size of vehicle is basically between 24 to 48 pixels. CWOC is manually annotated one class of positive samples ("car") with a number of 32,716, as well as four classes of negative samples ("boats", "trailers", "bushes" and "A/C units") that are easily confused with the vehicle with a number of 58,247. The annotation of each sample includes: sample class, and GT's center point coordinates.

4.3. DOTA

DOTA (Dataset for Object detection in Aerial images) is an aerial image dataset

made by Xia Guisong of Wuhan University, Bai Xiang of Huazhong University of Science and Technology, and others [11]. In order to eliminate the deviation caused by different sensors, the original material comes from multiple platforms (such as Google Earth). DOTA is characterized by multi-sensor and multi-resolution, namely that the GSDs of the images are diversified. DOTA contains a total of 2806 images about 4000 × 4000 pixels, and is manually annotated 15 classes of sample ("plane", "ship", "storage tank", "baseball diamond", "tennis court", "swimming pool", "ground track field", "harbor", "bridge", "large vehicle", "small vehicle", "helicopter", "roundabout", "soccer ball field" and "basketball court") with a number of 188,282.The annotation of each sample includes: sample class, and the coordinates of GT's 4 corners (where the top left corner is the starting point, arranged in a clockwise order).

5. A Vehicle Detection Method for Aerial Image Based on YOLO

In this paper, we process and integrate the above three public aerial image datasets first and then modify the network parameters of YOLO algorithm map propriately to train a model. Thus, we propose a vehicle detection method for aerial image. The specific steps are as follows.

5.1. Make Standard Datasets for YOLO Training

The standard dataset for YOLO training mainly consists of two parts: images and labels, where images are JPEG format and labels are txt format documents. Labels and images are in one-to-one correspondence. Each label records annotations of the samples in the corresponding image. The annotation format is:

class GT's center point coordinates (x, y) GT's width and height (w, h)

where (x, y, w, h) are normalized values, wrap the line to distinguish when there are multiple samples in one image. Since the input dimension of YOLO v3 training network is 416 × 416 × 3, the size of image used for training should not be too large, otherwise the characteristics of the sample after resize may be lost seriously. The basic information of the three public aerial image datasets described in Chapter 4 is shown in **Table 1**.

Table 1. The basic information of the three public aerial image datasets.

Dataset	Image Format	Images	Classes	Image size	Annotations
VEDAI	JPEG	1250	9	1024×1024	sample class, GT's center point coordinate, direction, coordinates of GT's 4 corners
COWC	TIFF	53	4	2000×2000 - 19,000 × 19,000	sample class, GT's center point coordinates
DOTA	JPEG	2,806	15	about 4000 × 4000	sample class, coordinates of GT's 4 corners

We process the above three datasets separately.

1) VEDAI

a) Image size is suitable and do not need to be processed;

b) Delete the annotation of "plane", "boat", and "other" three classes in labels;

c) Delete the "direction" in annotations;

d) According to the coordinates of GT's 4 corners, calculate width and height:

$$w = x_{\max} - x_{\min}, h = y_{\max} - y_{\min} \tag{1}$$

2) COWC

a) Delete the grayscale images;

b) Delete the annotation of negative samples, leaving only the positive sample "car";

c) Split the images of COWC into 416 × 416 size and convert to JPEG format. When splitting, the coordinate of the sample center point is converted accordingly to ensure its position in the new image is correct. The remaining images less than 416 × 416 are padded with black.

d) According to the GSD of COWC, it is assumed that the size of vehicle in the image is unified to 48 * 48 pixels, therefore,

$$w = h = 48/416 = 0.115384615384615... \tag{2}$$

3) DOTA

a) Except for "large vehicle" and "small vehicle", delete all the annotations of other 13 classes in labels, "large vehicle" and "small vehicle" are unified to "car";

b) Split the images of DOTA into 1024 × 1024 size. When splitting, the coordinates of GT's 4 corners are converted accordingly to ensure their positions in the new images are correct. Abandon the remaining imagesless than 1024 × 1024.

c) Center point coordinate:

$$x = (x_{\max} + x_{\min})/2, y = (y_{\max} + y_{\min})/2 \tag{3}$$

d) Width and height:

$$w = x_{\max} - x_{\min}, h = y_{\max} - y_{\min} \tag{4}$$

After processing, the information of the new datasets are shown in **Table 2**.

5.2. Configure Network Parameters for YOLO Training

1) Batch size

Use YOLO v3 default parameter $batch\_size = 64$.

Table 2. The processed datasets information.

Dataset	Image Format	Images	Classes	Image size	Annotations
VEDAI	JPEG	1250	6	1024 × 1024	$(class, x, y, w, h)$
COWC	JPEG	4944	1	416 × 416	$(class, x, y, w, h)$
DOTA	JPEG	14,348	1	1024 × 1024	$(class, x, y, w, h)$
Total	JPEG	20,542	6	416 × 416, 1024 × 1024	$(class, x, y, w, h)$

2) Number of iterations

The dataset contains a total of 20,542 images, so one epoch needs to iterate: $20542/64 \approx 320$ times.

The YOLO training defaults to iterate 160 epochs, so the number of iterations is: $160 \times 320 = 51200$ times.

3) Learning rate

The initial learning rate is 0.001, after 60 epoch divided by 10, after 90 epoch divided by 10 once again.

4) Number of filters in the last layer of the network

$$filters = (class + 5) \times 3 = (6 + 5) \times 3 = 33$$

6. Experimental Results

In this paper, we use NVIDIA's TITAN X graphics card for training. The training duration is about 60 hours. The test results of the training model are shown in **Table 3**.

The detection effect of the training model on unknown images are shown in **Figure 2** (the original images are from Internet, please inform if there is any infringement).

Figure 2 (left) shows that the training model has a good effect on detection of small objects. The vehicles in **Figure 2** (middle) are mostly not horizontal or vertical with rotation, test result shows that the model has a good performance on the detection of rotating objects, especially the leftmost vehicle in the image is very close to the background, while the manual detection may miss the object, and the model correctly detects it. **Figure 2** (right) indicates it is outstanding that the model on detection of compact and dense objects, more than 95% of the vehicles are correctly detected except for those in the far left shadow.

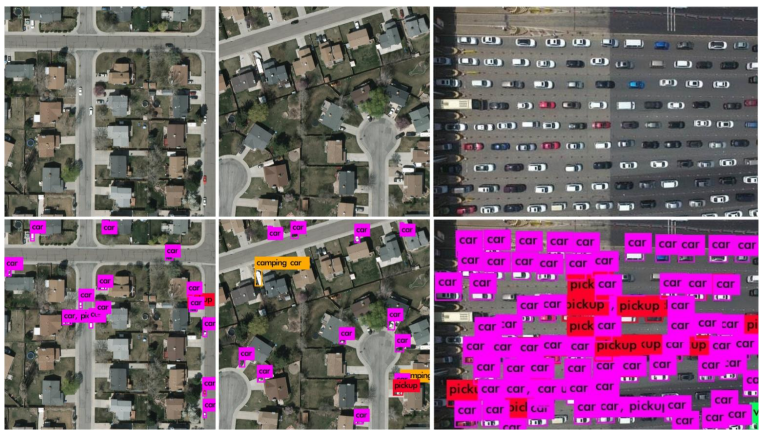

Figure 2. Training model test on unknown images.

Table 3. Test results of the training model.

Indicator	mAP	recall	IoU	fps
Value	76.7%	92.1%	82.3%	55

7. Conclusion

In this paper, a vehicle detection method based on YOLO deep learning algorithm for aerial image is presented. This method integrates an aerial image dataset suitable for YOLO training by processing three public datasets. The training model has good test results especially for small objects, rotating objects, as well as compact and dense objects, and meets the real-time requirements. Next, we will integrate more public aerial image datasets to increase the number and diversity of training samples, at the same time, optimize the YOLO algorithm to further improve the detection accuracy.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Qiu, Y. (2014) Video-Based Vehicle Detection in Intelligent Transportation System. Master Thesis, Jilin University, China.

[2] Cheng, P., Zhou, G. and Zheng, Z. (2009) Detecting and Counting Vehicles from Small Low-Cost UAV Images. *Proceedings of ASPRS* 2009 *Annual Conference*, Baltimore, 1-7.

[3] Azevedo, C.L., Cardoso, J.L., Ben-Akiva, M., Costeira, J.P. and Marques, M. (2014) Automatic Vehicle Trajectory Extraction by Aerial Remote Sensing. *Procedia-Social and Behavioral Sciences* (*S*1877-0428), **111**, 849-858. https://doi.org/10.1016/j.sbspro.2014.01.119

[4] Sivaraman, S. and Trivedi, M.M. (2010) A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking. *Nlgn Ranoraon Ym Ranaon on*, **2**, 267-276. https://doi.org/10.1109/TITS.2010.2040177

[5] Tehrani, H., Akihiro, T., Mita, S. and Mcallester, D.A. (2012) On-Road Multivehicle Tracking Using Deformable Object Model and Particle Filter with Improved Likelihood Estimation. *IEEE Transactions on Intelligent Transportation*, **2**, 748-758. https://doi.org/10.1109/TITS.2012.2187894

[6] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[7] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 6517-6525. https://doi.org/10.1109/CVPR.2017.690

[8] Redmon, J. and Farhadi, A. (2018) YOLO v3: An Incremental Improvement. arxiv:1804.02767v1 [cs.CV], Unpublished.

[9] Razakarivony, S. and Jurie, F. (2015) Vehicle Detection in Aerial Imagery: A Small Target Detection Benchmark. *Journal of Visual Communication & Image Representation*, **34**, 187-203. https://doi.org/10.1016/j.jvcir.2015.11.002

[10] Mundhenk, T.N., Konjevod, G., Sakla, W.A. and Boakye, K. (2016) A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning. *Proceedings of European Conference on Computer Vision*, Springer,

2016, 785-800.

[11] Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J.L., *et al*. (2018) DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. arXiv: 1711.10398v2 [cs.CV], Unpublished.