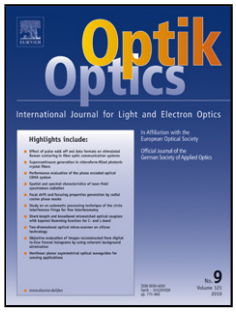

Title: An improved tiny-yolov3 pedestrian detection algorithm

Authors: Yi Zhang, Yongliang Shen, Jun Zhang

PII: S0030-4026(19)30155-X
DOI: https://doi.org/10.1016/j.ijleo.2019.02.038
Reference: IJLEO 62365

To appear in:

Received date: 7 January 2019
Accepted date: 12 February 2019

Please cite this article as: Yi Z, Yongliang S, Jun Z, An improved tiny-yolov3 pedestrian detection algorithm, *Optik* (2019), https://doi.org/10.1016/j.ijleo.2019.02.038

An improved tiny-yolov3 pedestrian detection algorithm

Yi Zhang[a], Yongliang Shen[a,*], Jun Zhang[a]

[a] College of Electronic Engineering, Heilongjiang University, Harbin 150080, China

[*]Corresponding author.

E-mail adress: 1995005@hlju.edu.cn

Abstract：The existing real-time pedestrian detection method often loses part of the detection accuracy. For this reason, we proposed a real-time pedestrian detection algorithm based on tiny-yolov3. The proposed method uses K-means clustering on our training set to find the best priors. We improved the network structure of tiny-yolov3 to make it more accurate in pedestrian detection. From the experimental results, the proposed method has higher detection accuracy under the premise of satisfying real-time performance.

Keywords：pedestrian detection; real-time; k-means; tiny-yolov3

1. Introduction

The vision system of pilotless automobile technology has always been a difficult point in the field of computer vision, and a reliable vision system will drive the development of pilotless automobile. In recent years, with the continuous enhancement of hardware computing power, deep learning based on convolutional neural networks [1] has developed rapidly, and achieved better achievement in the field of computer vision.

Currently commonly used methods are R-CNN series [2-5], SSD series [6, 7] and YOLO series [8-10]. The YOLO (You Only Look Once) algorithm proposed by Joseph Redmon and Ross Girshick, which solves the object detection as a regression problem and output the location and classification of the object on an end-to-end network in one step. The detection speed of this method is one of the fastest algorithms at present, but there is a large error in the detection accuracy of small targets, and it is impossible to accurately detect pedestrian targets in complex scenes. The YOLO algorithm has been constantly improved and the latest is YOLOv3, which uses the K-means clustering method to automatically select the best initial regression frame for the data set. The multi-scale anchor mechanism [11] is adopted to improve the detection accuracy of small objects.

W. Liu, D. Anguelov, et al. has proposed SSD (Single Shot MultiBox Detector) algorithm uses regression method for detection, and integrate positioning and classification into one network. The SSD network was modified on VGG16 [12] to replace the fully connected layer of VGG16 with the convolutional layer. Each convolutional layer that we added outputs a feature map and uses the feature map as an input to the prediction, resulting in a multi-scale

feature map for regression. The low-level feature map contains more information, which is good for retaining details, returning training errors, and improving the accuracy of detection. Compared with the YOLO algorithm, the recognition accuracy for small targets is improved.

All of the above methods only win various competitions (such as the COCO challenge [13]), and these methods are not applicable to the actual scene. For pilotless automobile, high real-time performance is required while the detection accuracy is required. These papers usually only indicate that they reach a certain frame rate, but do not fully understand the speed/accuracy trade-offs, depending on many other factors, such as which feature extractor to use, the input image size, and so on.

In this paper, we have a better trade-off between speed/accuracy and improved the tiny-yolov3 network to make it more suitable for unmanned driving.Tiny-yolov3 is a simplified version of YOLOv3, which has a much smaller number of convolution layers than YOLOv3, which means that tiny-yolov3 does not need to occupy a large amount of memory, reducing the need for hardware. And it also greatly speeds up detection, but lost some of the detection accuracy. In terms of the structure of the network we proposed, we deepen the network of tiny-yolov3, enhance the feature extraction ability of the objects, improve the detection accuracy. And we uses K-means clustering on our training set bounding boxes to find the best priors to make our model has better learning ability for our data sets.

2. Improved tiny-yolov3 network

The Tiny-yolov3 network is a network for detecting over 80 different object categories. The detection speed is the fastest algorithm at present, but the detection accuracy is very low compared to other algorithms. The object detection for complex scenes is not accurate enough. In the road scene, pedestrian objects are basically not detected. The algorithm we proposed in this paper is based on tiny-yolov3. We used K-means clustering to find the most suitable anchor boxes on our pedestrian dataset, and we deepened the backbone network of the tiny-yolov3 network so that the more semantic information can be extracted. The rich pedestrian features improve the detection accuracy while ensuring the detection speed, so as to meet the pedestrian detection in the unmanned road environment.

2.1 Selection of candidate frames based on K-means clustering

The anchor box mechanism has been proposed in Faster-RCNN and SSD. But the size of the prediction box is usually set manually, which will cause the network to converge slowly during training, and it is prone to local optimization. YOLOv2 draws on the anchor box mechanism of Faster-RCNN, but adopts the method of K-means clustering on the dataset to find the optimal number and size of anchor boxes.

The traditional K-means clustering method uses the Euclidean Distance function, but this means that larger boxes have more error clusters than smaller boxes, and the clustering results may deviate. To this end, we adopted IOU [14, 15] (the overlap ratio of the generated candidate box and the original marker box) score to evaluate the clustering result, thus avoiding the error caused by the scale of the box. The distance function can be computed as the following formula:

$$d\left(box, centriod\right) = 1 - IOU\left(box, centriod\right) \tag{1}$$

where box is the sample; $centriod$ is the center point of the cluster; $IOU(box, centriod)$ is the overlap ratio of the cluster box and the center box.

In this paper, we extracted the pedestrian image from the VOC dataset to form the pedestrian dataset. K-means clustering method is used to compare the IOU scores with different k values. We finally choose the K value of 6 considering the complexity of the model.

2.2 Improvements to the tiny-yolov3 backbone network

The backbone network of YOLOv3 uses the Darknet-53 network. The network is too complex, and requires very high computational power for hardware. Due to the complex structure of the network, the detection speed is also affected. Tiny-yolov3 is a simplified version of YOLOv3. The backbone network of tiny-yolov3 has only 7 convolutional layers and 6 pooling layers. The network structure of tiny-yolov3 is shown in Fig. 1. The simplified network improves the detection speed, but it also loses some of the detection accuracy [16]. The improved algorithm in this paper adds three convolutional layers based on the tiny-yolov3, and the deepened network can better extract pedestrian features and improve detection accuracy.

Although the deeper neural network can improve the detection accuracy, the model parameters in the deeper network model will also grow geometrically, which will greatly increase the calculation amount and occupy the memory resources. In the case of excessive computation, Resnet [17] proposed adding a 1*1 convolutional layer to reduce the amount of computation. In this paper, we draws on the method in Resnet, and introduces 1*1 convolution kernel in the network. On the one hand, it can reduces the computational complexity and saves memory resources. On the other hand, it also increases the nonlinear excitation function and improves the feature extraction ability of the network which can improve detection accuracy. The improved network structure is shown in Fig. 2:

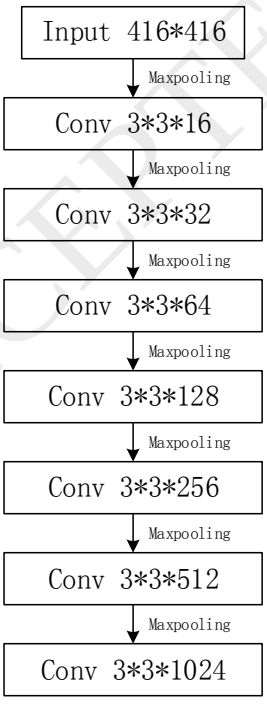

Figure 1. Network structure of tiny-yolov3.

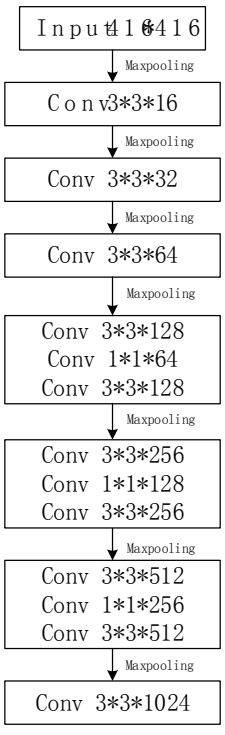

Figure 2. Network structure of the improved network.

2.3 Loss function

The YOLO algorithm is an end-to-end network, so the whole process uses a loss calculation called sum-square error [18], which is a simple addition of differences, including coordinate errors, IOU errors, and the classification error. The loss function can be expressed by the following formula:

$$loss = \sum_{i=0}^{S^2} coordErr + iouErr + clsErr \tag{2}$$

When the loss function is simply added, it is necessary to consider the weight of each loss function in the entire loss function. If the coordinate error is consistent with the classification error weight, the model will be unstable and will diverge during training. Therefore, the YOLO algorithm sets the weight of the coordinate error of $\lambda_{coord}=5$. When calculating the IOU error, the grid containing the object and the grid containing no object, the contribution of the IOU error to the network loss is different. If the same weight is used, the confidence value of the lattice containing no object is approximately 0, which distorts the influence of the confidence error of the lattice containing the object in calculating the gradient of the network parameter. To solve this problem, YOLO uses the $\lambda_{noperson}=0.5$ to fix the iouErr. (The 'contained' here means that there is an object whose center coordinates fall into the grid). For equal error values, the effect of large object errors on detection should be less than the effect of small object errors on detection. This is because the same positional deviation occupies a large object proportion much smaller than the equivalent position error occupies the proportion of small objects. YOLO solves this problem by square rooting the information items (w and h) of the object size, but it does not completely solve the problem.

Therefore, when training sample data, the overall loss function can be defined as:

$$\begin{aligned}
\text{loss} &= \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{person} \left[\left(x_i - x_i \right)^2 + \left(y_i - y_i \right)^2 \right] \\
&+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{person} \left[\left(\sqrt{w_i} - \sqrt{w_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{h_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{person} \left(C_i - C_i \right)^2 \\
&+ \lambda_{noperson} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{person} \left(C_i - C_i \right)^2 \\
&+ \sum_{i=0}^{S^2} l_i^{person} \sum_{c \in class} \left(P_i(c) - P_i\left(c\right) \right)^2
\end{aligned} \tag{3}$$

Where: S^2 is defined as the number of grids; B is defined as the number of prediction boxes

in each cell; (x, y) is defined as the center coordinates of each cell; w、h is defined as the width and height of the prediction box; C is defined as the confidence of the prediction box; P is defined as the confidence of the pedestrian; λ_{coord} is defined as the weight of the position loss function, the value is 5; $\lambda_{noperson}$ is defined as the weight of the classification loss function, the value is 0.5; l_{ij}^{person} is defined as whether there is a pedestrian object in the j-th prediction frame of the i-th cell, if there is a pedestrian target, the value is 1, otherwise 0 ; (x, y, w, h, C, P) is the corresponding predicted value.

The first line of the overall loss function uses the sum-squared error as the loss function for position prediction, and the second line uses the root-number error as the loss function for width and height, the third line and the fourth line uses SSE as the loss function for confidence, and the fifth row uses SSE as the loss function for the class probability. In the YOLOv3 algorithm, the IOU error loss function and the classification error loss function are calculated using binary classification cross entropy.

3. Experiments

The entire experimental platform configuration in this paper is shown in Table 1. The experimental environment compiles the entire script in Visual Studio 2015. The total number of iterations is 50020, the initial learning rate is set to 0.001, the learning rate is divided by 10 after 40,000 and 45,000 times, the mini-batch is set to 16, the subdivisions is set to 4, the weight attenuation coefficient is set to 0.0005, and the momentum coefficient is set to 0.9. During training, the input size of the model is changed 10 times per iteration, so that the final model has a better detection effect for images of different sizes.

3.1 Dataset

The PASCAL VOC 2007 [19] dataset contains a total of 20 categories of objects, a total of 9963 labeled images, providing a complete set of standardized and excellent data sets for image recognition and classification. The training method in this paper takes unsupervised learning and requires manual labeling of the data set. Considering that the manual annotation data set is a huge project, the data set of this paper directly extracts 4012 images of all pedestrians from the PASCAL VOC2007 data set. The background of the data set is complicated, the postures of the human beings are different, the degree of occlusion and the size of the human objects are not the same, which can enhance the generalization of the trained model and meet the complex traffic scenes of the road.

3.2 Experimental results and analysis

In order to evaluate the effectiveness of the proposed algorithm, this paper combines the above work, extracts the pedestrian image in the VOC2007 dataset as the pedestrian dataset, extracts 20% of the images as the test set, and draws the following PR curve [20] shown in Fig.

3 according to the recall rate and accuracy rate of the test results.:

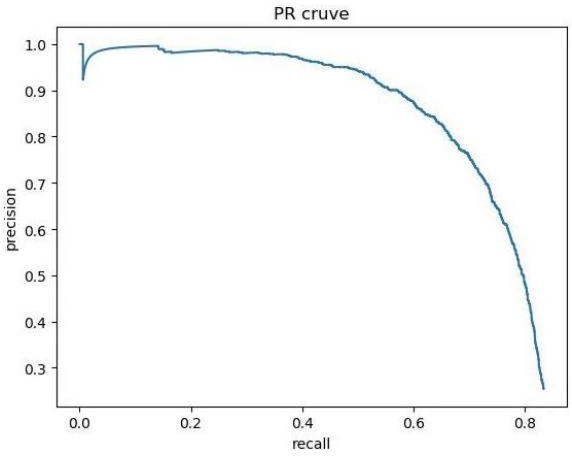

Figure 3. PR curve of improved network.

For comparison, this paper tested the AP value and detection speed of tiny-yolov2 and several advanced methods on the dataset of this paper. The experimental results are shown in Table 2:

Comparing the results of the above table, the method of this paper has achieved good performance, the AP value reached 73.98%, which is 5.44% higher than that of tiny-yolov3, which is more than that of tiny-yolov2. The proposed in this paper is effective for the improvement of the tiny-yolov3 network structure. For the detection speed, the proposed only needs 4.84ms to detect a picture, which is slightly higher than tiny-yolov2 and tiny-yolov3. This is because the algorithm of this paper deepens the network structure of tiny-yolov3, but it can also process 206 frames per second, which can meet the requirements of real-time.

The AP values of both large-scale network SSD and yolov3 algorithms reach above 80. The algorithm of this paper is lower than the two algorithms in detection accuracy, but the detection speed of these two large networks is slow, and a picture takes more than 20 milliseconds so that the real-time performance is not enough for pilotless automobile. Although the algorithm proposed in this paper loses part of the detection accuracy, it can meet the requirements of unmanned driving, and meet the requirements of real-time. The algorithm in this paper achieves a perfect balance between detection accuracy and detection speed, and is more suitable for pedestrian detection in real-time road environment.

In order to test the validity of the pedestrian detection algorithm more intuitively, we captures a set of real-time road video from the vehicle camera, and selects the 1543th image to test. The detection result is shown in Fig. 4:

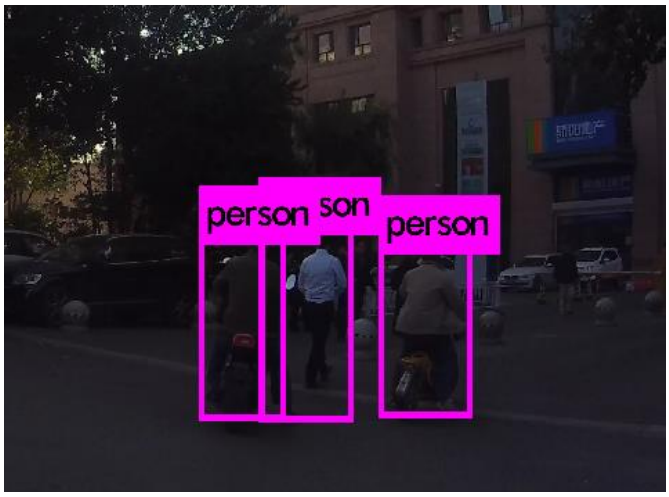

Figure 4. Detection effect of the improved network.

It can be seen that this article has a good detection effect for darker pictures, but tiny-yolov3 cannot detect pedestrians for darker pictures. This shows that the improved algorithm has a good adaptability to the complex real-time environment of the road, but we can also find that the algorithm in this paper still has missed detection, and the identification of small objects is not accurate enough, but the method we proposed has been greatly improved.

4. Conclusion

Based on the tiny-ylov3 network, we proposed an improved pedestrian detection network. For the different data sets, we use K-means clustering on our training set to find the best priors; in order to improve the network's ability to extract pedestrian features, we add three convolutional layers on the basis of the original network to improve the ability of the model to extract features; in order to reduce the amount of calculation caused by the deepening model, we introduce a 1*1 convolutional kernel to reduce the dimension of the feature, which greatly reduces the amount of calculation and ensures the real-time detection. The experimental results show that the proposed algorithm has higher accuracy for Pedestrian pictures in VOC2007, and has good robustness for real-time shooting of pedestrian pictures. However, the algorithm in this paper is still flawed. There are missed detections for smaller objects. How to improve the detection accuracy of smaller objects is the future research direction.

Acknowledgements

In the process of completing my paper, the gratitude would like to express to the professor Shen Yongliang for their great assistance. I would also like to thank my friends and classmates who gave me a lot of useful materials in the process of writing my paper, and also provided enthusiastic help in the process of typesetting and writing the paper.

References

[1] Lecun Y L ,Bottou L , Bengio Y , et al. Gradient-Based Learning Applied to Document Recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[2] Girshick R. Fast r-cnn[C]. Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[3] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[4] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]. Advances in neural information processing systems. 2015: 91-99.

[5] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]. Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017: 2980-2988.

[6] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector [C]. European conference on computer vision. Springer, Cham, 2016: 21-37.

[7] Fu C Y, Liu W, Ranga A, et al. DSSD: Deconvolutional single shot detector [J]. arXiv preprint arXiv:1701.06659, 2017.

[8] Redmon J, 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. 2016: 779-788.

[9] Redmon J, Farhadi A. YOLO9000: better, faster, stronger [J]. arXiv preprint, 2017.

[10] Redmon J, Farhadi A. Yolov3: An incremental improvement [J]. arXiv preprint arXiv:1804.02767, 2018.

[11] Erhan D, Szegedy C, Toshev A, et al. Scalable object detection using deep neural networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 2147-2154.

[12] SimonyanK, Zisserman A . Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer Science, 2014.

[13] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]. European conference on computer vision. Springer, Cham, 2014: 740-755.

[14] NowozinS . Optimal Decisions from Probabilistic Models: The Intersection-over-Union Case[C]. IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2014.

[15] Ahmed F ,Tarlow D , Batra D. Optimizing Expected Intersection-Over-Union with Candidate-Constrained CRFs[C]. IEEE International Conference on Computer Vision. IEEE, 2016.

[16] Huang J, Rathod V, Sun C, et al. Speed/accuracy trade-offs for modern convolutional object detectors[C]. IEEE CVPR. 2017, 4.

[17] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[18] Ranjbar M, Mori G, Yang W. Optimizing Complex Loss Functions in Structured Prediction[C]. European Conference on Computer Vision. 2010.

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, andA. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV, 2010.

[20] Erhan D, Szegedy C, Toshev A, et al. Scalable object detection using deep neural networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 2147-2154.

Table 1 Experimental platform configuration.

Names	Related configuration
operating system	Windows
CPU /GHz	Inter CoreI7-8700K，3.7
RAM /GB	16
GPU	NVIDIA GeForce GTX 1070，8
GPU acceleration library	CUDA10.0，CUDNN7.4

Table 2 Comparison of various algorithm test results.

Detection algorithm	AP value	Detection speed
Tiny-yolov2	63.88	4.76ms
Tiny-yolov3	68.54	4.81ms
improved network	73.98	4.84ms