The Practical Application of the YOLOv3 Model from the Perspective of Machine Learning in Pedestrian Detection

Yanan Li¹,a, Xuehong Sun²,b*, Liping Liu³,c

¹School of Physics and Electrical Engineering, Ningxia University Yinchuan, china
²School of Information Engineering, Ningxia University Yinchuan, china
³School of Physics and Electrical Engineering, Ningxia University Yinchuan, china

b*sunxh@nxu.edu.cn
a liyanan9410@163.com, b nxsunxh@163.com, c liuliping8186@126.com

Abstract—Target detection and tracking technology is particularly important in the field of computer vision. With the needs of social production and the development of science and technology, the research of video pedestrian detection and tracking technology based on mobile platforms also needs to be put on the agenda. In this paper, the mobile robot is based on the Gmapping method to map the unknown environment, based on the A* and Dijkstra algorithm to complete the safety inspection in the known environment, collect, record and save the video during the inspection process, and based on YOLOv3, the pedestrian target in the video for detection, the Deep_Sort target tracking algorithm establishes tracking of the detected target, and creates a learning sample data set 2019.

1. INTRODUCTION
The research on the detection and tracking technology of moving targets has gradually become an important research direction in the field of security in the process of social development. Pedestrian detection is actually to separate pedestrians from the background information of the image, and then determine the position of the target in the image, and it is customary to mark the detected target with a rectangular frame.

At present, common pedestrian detection technologies mainly include target detection algorithms based on area recommendations (R-CNN, etc.) and target detection algorithms based on regression methods. The former generates candidate frames first, and then classifies the samples through the convolutional neural network. This method has high detection accuracy, but it takes a long time and the detection speed will slow down. The latter does not need to generate candidate frames, and converts the target frame positioning problem into a regression problem. This method has a fast detection speed, but the detection accuracy is relatively low.

The YOLO series methods that have appeared in recent years can take into account the dual requirements of accuracy and speed in pedestrian detection. Therefore, this article chooses the YOLOv3 algorithm with a better comprehensive performance in terms of speed and accuracy as the prototype of the detection network. By comparison, it can be found that the tracking effect of Deep_Sort is better than the Sort method. Through the optimization processing of the selected method, the pedestrian is detected and tracked within the designated mobile robot navigation range. The experimental test results show the completeness, efficiency and rationality of the method proposed in this paper.
2. Pedestrian Detection

2.1. Collect learning samples
To study pedestrian detection technology, there must be suitable learning samples for the learning, training and testing of network models. The current online data sets contain many types, such as VOC2007. Therefore, in the research of this article, the 2019 data set that meets the requirements (only for pedestrians to be detected, thereby improving the detection accuracy) is specially produced as this time study samples for research. The production process mainly includes the acquisition of color images, the annotation of pedestrian targets in the image, and the production of data sets.

In the process of color image collection, select a video point with a clear view and a better scene on the satellite map as the point for acquiring the video stream, and determine the RTSP address corresponding to the selected video point camera. Download and install the VLC media player, then open the network stream and enter the RTSP address. After confirming that the RTSP address is correct, download the real-time video at that point. Then get each frame of the video from the local.

Use labelimage labeling software to frame and label each pedestrian in each frame of the image obtained in the previous step.

In order to meet the needs of the network model data format and the detection accuracy of pedestrian targets, this paper converts the marked 2019 data set into the coordinate format required by the network model. Specifically, it includes three parts: adjusting and renaming all images, making the 2019 data set in VOC format, and converting VOC to YOLO.

2.2. YOLOv3 target detection method
The YOLO algorithm \cite{1} is a deep neural network model for object detection. It is different from the two-step detection algorithm. The feature extraction, candidate frame regression and classification of the YOLO series are completed in the same branchless convolutional network. Makes the network structure simpler.

The basic process of detection is: divide the picture into grids, then predict a number of rectangular boxes for each grid, and finally judge whether there are objects in the rectangular boxes to complete the detection work.

The backbone network in YOLOv3 is mainly a 53-layer convolutional neural network, named darknet-53, and draws on the feature pyramid network \cite{2} to design a multi-feature extraction structure. The basic structure of the model is shown in Fig. 1:

![YOLOv3 model structure](image)

Fig. 1 YOLOv3 model structure

The Darknet-53 feature extraction network and the multi-scale fusion feature network together form a YOLOv3 structure network with three different-scale feature map outputs and use multiple small-size convolutional layers, thereby enhancing the robustness of the network.

2.3. Pedestrian detection method based on YOLOv3
In order to reduce the false detection of the YOLOv3 network model in the process of prediction, the prediction boxes are sorted according to the confidence level from high to low, and the intersection ratio is calculated most reliably according to the following prediction boxes. When it is greater than the threshold 0.45, it has Forecast boxes with low confidence will be deleted. After traversing and querying all the boxes, repeat the previous operation for the second prediction box in the remaining boxes until all the repeated boxes are eliminated, and finally obtain the coordinate information with the highest confidence and the probability of the category.
By increasing the number of anchor boxes, the problem of missing detection can be effectively solved. Therefore, K-means++ [6] clustering is used to cluster the width and height of the anchor boxes. The steps are as follows:

1. Arbitrarily select the width and height of a target frame as the first cluster center;
2. The box with the larger similarity distance to the current n−1 cluster centers is selected as the nth cluster center;
3. Repeat the operation of the previous step to determine all cluster centers;
4. Alternately calculate the intersection ratio between the cluster center and the remaining coordinates, and calculate the distance intersection ratio loss value between the two boxes;
5. After traversing each coordinate box, calculate the average value of the width and height of the coordinate box in each category as the center of the next iteration group;
6. Repeat (4) and (5) until the total intersection ratio difference of adjacent iterations is less than a preset threshold, and the clustering algorithm stops.

On the training set in the VOC2007 data set and the 2019 data set established in this article, the width and height of the four anchor boxes of 9, 12, 15, and 18 are respectively clustered. The anchor boxes are evenly distributed on the three output feature layers to obtain different model of the number of anchor frames and training on these two data sets.

In order to test the different effects of different sizes of input images on the detection, the network 9 groups of different input image sizes are set up to cluster 9 anchor box widths and heights for different sizes and train on the 2019 data set.

According to the picture size in the 2019 data set, the size of the network input is adjusted to 544*320.

For the datasets VOC and COCO for training and testing of YOLOv3, there are many types of images included in these two types of datasets, which affect the detection accuracy of a specific target. Therefore, 16420 untransformed images are specially produced. The pedestrian detection data set 2019 is used to complete the pedestrian detection work in this paper. In order to test the efficiency of this data set, it is specially used to train and test YOLOv3, and compare it with the detection of the VOC2007 data set.

When the VOC2007 data set is used as the input image of the original YOLOv3 network model, the VOC2007_train training network, VOC2007_test and 2019_test test network are selected.

In addition, in order to obtain the influence of the detectable multi-category network model on a single detection time, the relevant parameters in the yolo layer in the YOLOv3 network structure are modified and adjusted. Tested in the VOC2007 data set, the test results show that the detection accuracy has slightly changed, but it is not obvious.

Finally, a single pedestrian detection YOLOv3 network is trained and tested with the 2019 pedestrian detection data set containing only pedestrians.

3. EXPERIMENTAL ANALYSIS

In this experiment, the EMI Dashgo mobile robot patrols and records the surveillance video in the designated area as the test object, and detects and tracks the pedestrians in it.

The mobile robot uses G4’s high-precision lidar to scan 360 degrees without blind spots and omni-directional scanning combined with the Gmapping mapping method to scan and build maps in the designated area and save them. The experimental test results are shown in Fig. 2:
On the basis of the existing maps, the typical A* and D* algorithms in the global path planning algorithm combined with the teb local path planning algorithm are used to realize the navigation task of the mobile robot.

After mapping and navigation, mobile robots can already use G4's high-precision lidar and ultrasonic to avoid obstacles during walking. The decisive role in the obstacle avoidance process is the G4's high-precision lidar, so the position of the radar determines the height at which the mobile robot can avoid obstacles. On this basis, the mobile robot can basically perform inspections within a short distance within the specified range on the built map.

The experimental results of increasing the number of anchor frames to solve the loss detection problem are shown in Table 1:

| Training set   | Test set   | Bflops | Ap%  | Anchor box |
|----------------|------------|--------|------|------------|
| VOC2007_train | VOC2007_test | 51.62  | 62.36| 9          |
|                |            | 52.34  | 62.48| 12         |
|                |            | 53.51  | 62.49| 15         |
|                |            | 54.02  | 62.49| 18         |
| 2019_train     | 2019_test  | 53.41  | 64.21| 9          |
|                |            | 54.13  | 64.30| 12         |
|                |            | 55.35  | 64.34| 15         |
|                |            | 56.02  | 64.35| 18         |

It can be seen from the table that the number of anchor frames on the output feature layer has been increased, and the detection accuracy of the network model has been improved to a certain extent. It should be pointed out that increasing the number of anchor frames does not change the structure of the network, but only changes the dimensions of each output layer, so the detection speed of the network model has not changed.

Besides, the effect of input image size on detection speed and detection accuracy. The experimental results are shown in the table 2:

| Training set   | Test set   | Bflops | Ap%  | Input picturesize |
|----------------|------------|--------|------|-------------------|
| 2019_train     | 2019_test  | 48.64  | 43.36| 320*320           |
|                |            | 56.75  | 46.52| 352*352           |
|                |            | 65.63  | 49.80| 384*384           |
Take the picture size data in the table as the abscissa, and use the floating point number and the model detection accuracy as the ordinate to draw a graph of the floating point number and the detection accuracy varying with the input picture size, as shown in Fig.3:

![Graph showing model calculation and detection accuracy varying with the size of the picture](image)

Fig.3 Model calculation and detection accuracy vary with the size of the picture

It can be seen from the graph that increasing the size of the network picture can improve the detection accuracy of the network model, but the amount of network floating point calculations will also increase.

The amount of floating-point calculations depends on the output size of the convolutional layer. The size of the input image has a greater impact on the amount of floating-point calculations of the network and will reduce the speed of model calculations. Therefore, when choosing the size of the input model of the network model, it is necessary to ensure both accuracy and speed.

And use 2019_test to test the model, and the test results As shown in Fig.4.

![Image of pedestrian detection network test based on YOLOv3 2019_test](image)

Fig.4 Pedestrian detection network test based on YOLOv3 2019_test

In summary, the comparison results between the original YOLOv3 target detection network and the pedestrian detection network are shown in Table 3:

| data set  | index         | Original YOLOv3 | Pedestrian detection YOLOv3 |
|-----------|---------------|-----------------|-----------------------------|
| VOC 2007  | detection accuracy% | 89.9            | 95.6                        |
|           | missed detection rate% | 0.28            | 0.15                        |
|           | false detection rate%  | 9.82            | 4.25                        |
In the pedestrian detection network based on YOLOv3, the training test contains only the pedestrian data set 2019, and the detection accuracy is significantly better than the original YOLOv3 target detection network; when the public data set VOC2007 is used as the data set, the pedestrian detection network based on YOLOv3 has the detection accuracy of pedestrians Higher than the original YOLOv3. Therefore, the effectiveness of this method in improving the accuracy of pedestrian detection can be obtained.

In addition, for the problem of low recognition rate of difficult samples (scenes, poor light, small targets, etc.), the number of difficult samples was increased in the experiment to train a dedicated pedestrian detection YOLOv3 network model. It shows that this method has a certain effect on improving the recognition rate of difficult samples.

Finally use the obtained video as data for subsequent test experiments. Choose a video of intermediate quality (light, pedestrian number, occlusion, etc.) as the test set. In the configured Numpy, sklearn, OpenCV, and TensorFlow-gpu environments, download the trained yolov3.weights and convert the weights to the yolo.h5 model. Finally, modify the path of the test video in demo.py to complete the test.

In this experiment, a 30-second short video in the intercepted MOT data set \cite{4} was selected as the input. The video included a total of 32 pedestrian targets. The test results are shown in Fig.5 and Table 4.

![Fig.5 Test result](image)

| data   | index   | Original YOLOv3 | Pedestrian detection YOLOv3 |
|--------|---------|----------------|-----------------------------|
| MOT    | successful match% | 86.4            | 93.6                        |
|        | mismatch%       | 13.6            | 6.3                         |

According to the chart, without changing the tracking model, the original YOLOv3 network and the dedicated pedestrian detection YOLOv3 network have obvious differences in the performance of pedestrian detection and tracking in this experiment. The results show that the dedicated line adjusted based on this article The human detection YOLOv3 network is the detection model, and the network model that introduces Deep-Sort as the tracking model can successfully detect and match most target pedestrians in this experiment, which proves the feasibility of this method.

4. CONCLUSION
Target detection and tracking have different methods and technologies in different scenarios. This article is only aimed at detecting pedestrians and tracking motion trajectories under the general
background of specific pedestrian detection, so as to meet the requirements of pedestrian detection and tracking in security surveillance videos.

ACKNOWLEDGMENT
Thanks to the instructor, Professor Sun Xuehong, for his follow-up and good guidance, and thanks to the Ningxia Natural Science Foundation key project funding (Item Number: 2020AAC02006).

REFERENCES
[1] Redmon J, Farhadi A. Yolov3: An incremental improvement[J/OL]. https://arxiv.org/abs/1804.02767. [2018-04-08].
[2] Shaoqing Ren, Kaiming He, Ross Girshick, ang Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Neural Information Processing Systems (NIPS), 2015.
[3] Sermanet P, Kavukcuoglu, Chintala S, et al. Pedestrian detection with unsupervised multistage feature learning. Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013. 3626-3633.
[4] Bay H, Tuytelaars T, Van Gool L. SURF: speeded up robust features. In: Proceeding of Computer Vision-ECCV2006. Lecture Notes in Computer Science, vol. 3951. Berlin, Heidelberg: Germany Springer, 2006. 404-417
[5] Sermanet P, Kavukcuoglu, Chintala S, et al. Pedestrian detection with unsupervised multistage feature learning. Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013. 3626-3633.
[6] u B, Ju FY. An optimized genetic K-means clustering algorithm. Proceedings of 2012 International Conference on Computer Science and Information Processing. Xi’an, China. 2012. 1296-1299.