Algorithm of Pedestrian Detection Based on YOLOv4

Qinjun Zhao[1(✉)], Kehua Du[1], Hang Yu[1], Shijian Hu[1], Rongyao Jing[1], and Xiaoqiang Wen[2]

[1] School of Electrical Engineering, University of Jinan, Jinan 250024, China
cse_zhaoqj@ujn.edu.cn

[2] Department of Automation, Northeast Electric Power University, Jilin 132012, China

Abstract. Pedestrian detection technology is applied to more and more scenes, which shows high application value. In recent years, with the development of electronic information technology, the computing speed of computers has been growing rapidly, and the deep learning technology has become better and better with the development of computers. In this paper, based on YOLOv4, this paper studied the scheme of pedestrian detection, obtained the anchor of the pedestrian data through the K-Means algorithm, the loss function of the target detection algorithm is optimized, and introduced the Soft-NMS to improve the pedestrian occlusion problem in detection. Through relevant model verification experiments, the algorithm in this paper is faster than the traditional target detection algorithm in terms of speed, accuracy and robustness.

Keywords: Deep learning · Pedestrian detection · YOLOv4 · Soft-NMS

1 Introduction

At present, pedestrian detection has been applied in many aspects, such as assisting the intelligent driving of cars, monitoring the safety of pedestrians in important places, providing a basis for pedestrian behavior analysis, pedestrian number statistics or pedestrian tracking in video surveillance. In various practical scenarios, the detection of pedestrians through computers has greatly improved work efficiency, reduced labor costs, and promoted social and economic progress and development.

At present, pedestrian detection methods mainly include two kinds: traditional vision method and the depth learning method. The traditional way of vision is to manually analyze the characteristics of pedestrians, extract the pedestrian foreground, and then design a classifier for discriminant analysis. The typical ones include Haar feature detection, HOG feature detection, and DPM feature detection [1–3]. The accuracy, speed and robustness of traditional detection models all have certain defects. With the continuous improvement of computing power of computer chips, at present, the deep learning algorithm using convolutional neural networks has been favored by researchers, and have achieved excellent results in various image detection fields. YOLO (You Only Look

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 535, pp. 231–239, 2024.
https://doi.org/10.1007/978-3-031-50580-5_19

Once) as the representative of the one-stage detection network, the peer-to-peer design is realized, and the algorithm shows good detection effect and robustness in various application scenarios through training on specific image data [4–7]. This paper studies a set of algorithms for pedestrian detection based on the YOLOv4[7] algorithm, and optimizes and improves the training strategy.

2 YOLOv4

With the introduction of the first generation of YOLO algorithm, its rapidity and accuracy have been continuously loved by researchers and applied to various detection fields. After years of exploration, various versions of YOLO's derivative structures have been continuously proposed, and the most classic version is YOLOv3[6]. The YOLOv4 target detection algorithm is also an improvement of YOLOv3 detection algorithm. The pedestrian detection algorithm proposed in this paper is also an improvement of YOLOv4 target detection algorithm.

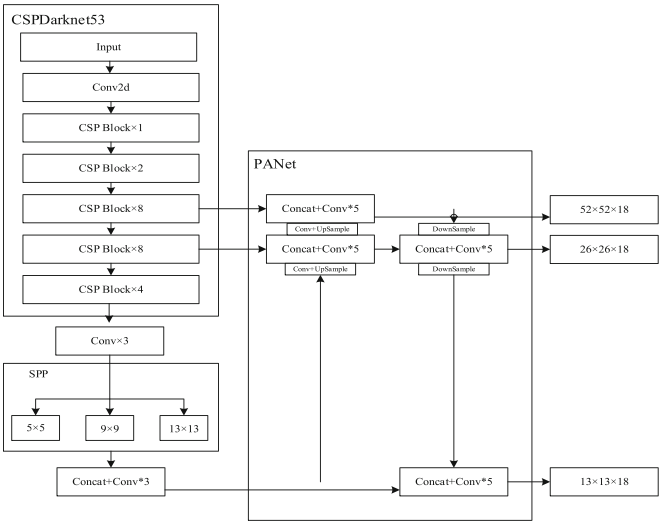

Fig. 1. YOLOv4 structure

A complete convolutional neural network mainly has two calculation stages. The primary stages is called the 'Backbone', which is mainly used for image convolution to obtain image features, which is the foundation of the network; the second part is the 'Neck' part after Backbone, which is used for other tasks (object classification, semantic segmentation, etc.).

The content in Fig. 1 is the composition of YOLOv4. In view of the network degradation problem that may occur in the neural network, the backbone network uses the CSPParknet53 structure, which contains multiple CSP residual block structures [8]. The neck contains the PANet structure and SPP pooling module. In order to achieve better

performance of network accuracy, the SPP pooling module uses 5x5, 9x9 and 13x13 pooling layers for feature fusion. The PANet module can accurately retain spatial information and help locate the target through the fusion of upper and lower layer features. The YOLOv4 network finally has three branch outputs, which are respectively generated as matrix grids of 13×13, 26×26, and 52×52. It provides different sizes of receptive fields for three types of targets, large, medium and small. Each grid predicts 3 prediction boxes, each prediction box contains three types of information: object confidence, classification, and position (x, y, w, h). If only the single type of pedestrian target is detected, a total of $3 \times 6 = 18$ numbers.

3 Loss Function

In CNN training, the loss function plays an important role in the iterative process. By calculating its value and then feeding it back to the network, the network parameters can be corrected. Some loss function formulas mentioned in this paper are as follows:

$$l_{box} = \sum\nolimits_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \left(1 - IoU\left(A^{pre}, B^{gt}\right)\right) \tag{1}$$

$$l_{cls} = \sum\nolimits_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{obj} BCE\left(\widehat{P}(c_i), P(c_i)\right) \tag{2}$$

$$l_{obj} = \sum\nolimits_{i=0}^{S^2} \sum_{j=0}^{B} (1_{i,j}^{obj} BCE\left(\widehat{c_i}, c_i\right) + 1_{i,j}^{noobj} BCE\left(\widehat{c_i}, c_i\right)) \tag{3}$$

Among them, S^2 refers to the number of grids, B indicates the number of prediction boxes contained in the grid, $1_{i,j}^{obj}$ represents that the grid contains targets, and $1_{i,j}^{noobj}$ represents that there are no targets in the grid. Since the earliest YOLO algorithm, the loss function has been playing an important role in calculating the network parameter optimization network. At the beginning, use the square difference loss function to calculate the network variables. At present, the loss function used by the network has been replaced by IoU loss function based on anchor points. Compared with the former, this loss function also has many positive effects on the accuracy of the model and the training speed of the model. Another loss function is used for object and classification loss, which is called cross entropy loss function.

3.1 Bounding Box Loss

Correct the bounding box by adjusting the preset Anchor point to fit the target. Before training, the collected pedestrian width and height data are used as a bunch of discrete data, then classify the data, which is generally completed by K-Means algorithm, they are: (10×24), (18×46), (27×104), (40×64), (56×148), (85×265), (121×174), (166×312), (303×371).

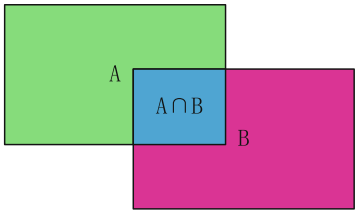

Fig. 2. IoU schematic

Figure 2 shows the relationship between prediction box A with target box B, and the formula for calculating IoU is:

$$IoU = \frac{A \cap B}{A \cup B} \tag{4}$$

When the two boxes are far apart, the IoU value is 0, and the IoU value at this time cannot represent the relationship between the two boxes, resulting in the phenomenon that the loss unable to make effective corrections to the CNN network. In response to this problem, the paper [9] proposed the GIoU algorithm, which considered the proportions of A and B in the complement part of the minimum enclosing matrix. The paper [10] proposes DIoU and CIoU algorithms, The former increases or decreases the square of the distance between the diagonals of the new rectangular area formed by the intersection of two frames according to IoU loss, and the latter adds aspect ratio to DIoU. Combining the ideas of the above algorithms, in this paper, some relevant calculation formulas are summarized to express the relationship between prediction box, target box and boundary box. The loss calculation formula used is as follows:

$$\Delta = \left| \frac{h^{pre}}{h^{pre} + w^{pre}} - \frac{h^{gt}}{h^{gt} + w^{gt}} \right| \tag{5}$$

$$l_{IoU} = 1 - IoU(A, B) + \frac{\rho^2\left(b^{pre}, b^{gt}\right)}{c^2} * e^{\alpha\Delta} \tag{6}$$

where Δ indicates the difference between the absolute values of the length ratio and width ratio between two boxes, $\frac{h}{h+w}$ in the range of (0,1). ρ^2 is the second power of the length of the connecting line between the geometric center of the prediction box and the geometric center of the target box, c^2 is the square of the distance between the diagonals of the coincident parts of two boxes. $e^{\alpha\Delta}$ is added as a coefficient correction, when $\Delta \to 0$, $e^{\alpha\Delta} \to 1$, the selection of α has different influence on the model fitting speed. In this algorithm, $\alpha = 10$ is selected to enhance the fitting speed of the model. It is shown in the Fig. 3 that influence of different IoU schemes on AP (Average Precision) in cross-validation set.

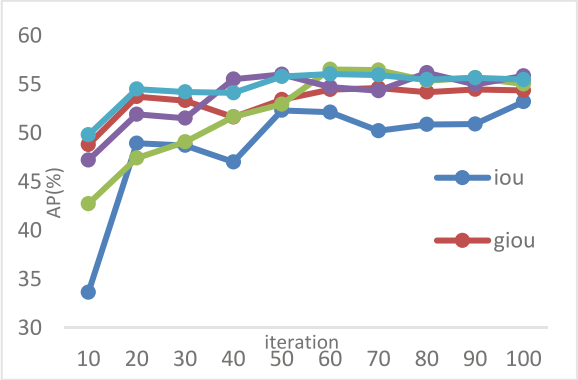

Fig. 3. Comparison of effects of different IOU schemes

3.2 Bounding Box Loss

BCE loss function is used for object and class loss:

$$BCE(\hat{y},y) = -[ylog\hat{y} + (1-y)log(1-\hat{y})] \tag{7}$$

In the actual training process, YOLOv4 divides the picture into $N \times N$ regions, and each region detects whether there is a target. Since the data set used in this paper contains fewer samples of pedestrians, the number of samples containing background information is larger, and the small number of pedestrians in a single picture. In calculating the object loss of a single image, the proportion of negative samples will be too large, causing too much loss weight to be biased towards background samples. Therefore, the new loss function is adopted in this paper to adjust negative samples through a variable γ coefficient:

$$l_{obj} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (1_{i,j}^{obj} BCE(\hat{c_i},c_i) + 1_{i,j}^{noobj} BCE(\hat{c_i},c_i) * \gamma) \tag{8}$$

γ is affected by the iteration rate x, which is small at the beginning, and gradually returns to the normal state according to the progress of the iteration (Fig. 4):

$$\gamma = \sqrt{\sin\left(\frac{\pi}{2}x\right)}, x \in (0, 1] \tag{9}$$

By using the γ coefficient, the loss converges faster and the curve is smoother, as shown in Fig. 5.

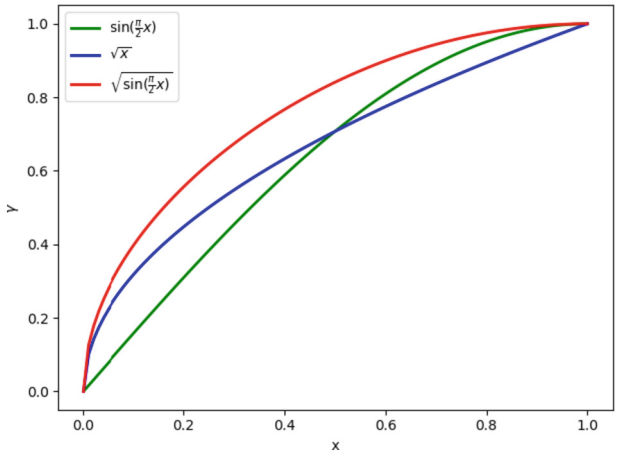

Fig. 4. Smoother γ graph, iteration rate x from 0 to 1

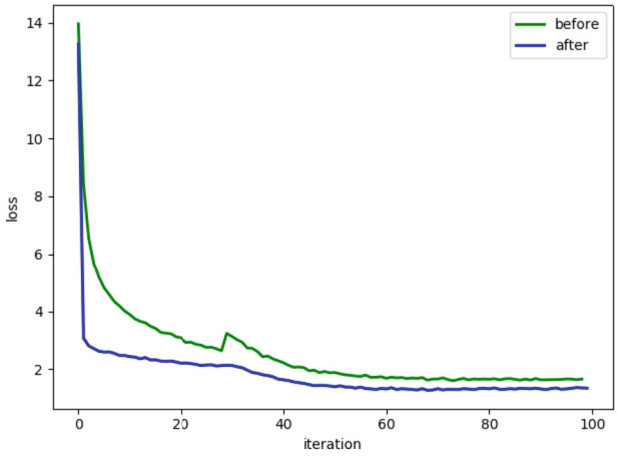

Fig. 5. Changed loss graph

4 Soft-NMS

Since the deep learning network will generate multiple prediction boxes for the target in the input image, the Non-Maximum Suppression (NMS) algorithm will sort the prediction boxes with high IoU coincidence by their confidence, delete the prediction boxes with lower confidence, and finally keep only one prediction box. This algorithm is simple and efficient, but it also has certain drawbacks. When two or more targets are close to each other in the image, using the NMS algorithm in the model may result in some targets not being detected, resulting in lower detection accuracy. This paper combines the Soft-NMS algorithm proposed in the paper [11] to improve this problem:

$$s_i = \begin{cases} s_i, & iou(M, b_i) < N_i \\ s_i(1 - iou(M, b_i)), & iou(M, b_i) \geq N_i \end{cases} \tag{10}$$

In the face of prediction boxes with high coincidence, Soft-NMS adopts a strategy of reducing the score of low-confidence prediction boxes, which is different from the traditional NMS algorithm that directly deletes low-confidence prediction boxes (Fig. 6).

By introducing the Soft-NMS strategy, to some extent, can improve the detection effect, the actual detection effect is shown in Fig. 7.

Input : $\mathcal{B} = \{b_1, .., b_N\}$, $\mathcal{S} = \{s_1, .., s_N\}$, N_t
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
N_t is the NMS threshold

begin
  $\mathcal{D} \leftarrow \{\}$
  while $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow$ argmax $\mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{M}$; $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
    for b_i *in* $\mathcal{B}$ **do**
      if $iou(\mathcal{M}, b_i) \geq N_t$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_i$; $\mathcal{S} \leftarrow \mathcal{S} - s_i$
      end NMS
      $s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$ Soft-NMS
    end
  end
  return $\mathcal{D}, \mathcal{S}$
end

Fig. 6. Pseudo code of Soft-NMS replacing NMS

Fig. 7. Soft-NMS improves results

5 Experimental Results and Analysis

The videos taken at different locations in the school are used for video data in this paper, including two environments, day and night. In order to facilitate processing, the data size is uniformly compressed to 720 × 406 in proportion. The pictures used for training and testing are sampled every 3 frames in the video, only to retain higher quality data. Select some images containing "person" in the PASCAL VOC dataset [12] to enrich the dataset samples. In this paper, 10903 images are taken from 12114 images in the final data set as the training set of the model, and 1211 images are taken as the verification set of the model. The equipment used for training the model is i9-9900X CPU, two RTX 2080 graphics cards with 24G video memory, and the equipment used for testing is i5-9300H CPU, one GTX 1660 graphics card with 6G video memory, and the systems used for training and testing are all Ubuntu operating systems. The initial training was based on the YOLOv4 pre-training weights, finally, verify the model on the verification set, and retain the model with the best performance. This paper summarizes the data from the model test on the test set to Table 1. By analyzing the data of various performance indicators in the table, it can be found that the detection effect and detection speed of the traditional visual algorithm lag behind YOLOv4 no matter in the day or at night. To sum up, the algorithm in this paper can meet the needs of high real-time applications due to its fast detection speed (Fig. 8).

Table 1. Test Results

Model	Detect accuracy(day)	Detect accuracy(night)	FPS
DPM + SVM	80.41%	72.81%	11.52
YOLOv4(ours)	92.18%	83.46%	25.18

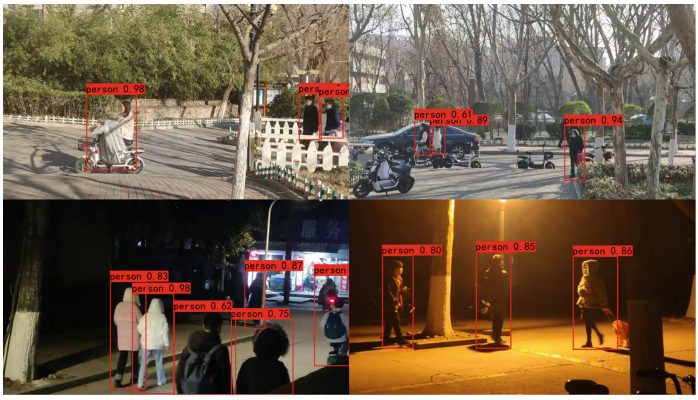

Fig. 8. Part of the detection effect

Through the experiments on traditional visual algorithms and the depth learning algorithm based on YOLOv4, we can draw the following conclusions: the pedestrian

detection algorithm based on depth learning is superior to the traditional visual algorithm in terms of detection accuracy, detection speed and detection robustness. It is true that deep learning algorithms perform well in all aspects, but the algorithms also have shortcomings such as dependence on data sets and long training time. It is believed that in the future, the target detection model based on deep learning will have higher accuracy and faster speed, and will play an increasingly important role in various detection tasks.

Acknowledgement. This work is supported by the Shandong Key Technology R&D Program 2019JZZY021005, Natural Science Foundation of Shandong ZR2020MF067, ZR2021MF074 and ZR2022MF296.

References

1. Sumit, S.S., Rambli, D., Mirjalili, S.: Vision-based human detection techniques: a descriptive review. IEEE Access **9**, 42724–42761 (2021)
2. Cheng, E.J., Prasad, M., Yang, J., et al.: A fast fused part-based model with new deep feature for pedestrian detection and security monitoring. Measurement **151**, 107081 (2020)
3. Khemmar, R., Delong, L., Decoux, B.: Real time pedestrian detection-based Faster HOG/DPM and deep learning approaches. Inter. J. Comput. Appli. **176**(42), 34–38 (2020)
4. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788. IEEE, Las Vegas (2016)
5. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 6517–6525. IEEE, Hawaii (2017)
6. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv 1804, 02767 (2018)
7. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv 2004,10934 (2020)
8. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE, Las Vegas (2016)
9. Rezatofighi, H., Tsoi, N., Gwak, J.Y., et al.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666. IEEE, Long Beach (2019)
10. Zheng, Z., Wang, P., Liu, W., et al.: Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12993–13000. AAAI, New York (2020)
11. Bodla, N., Singh, B., Chellappa, R., et al.: Soft-NMS--improving object detection with one line of code. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5561–5569. IEEE, Venice (2017)
12. Everingham, M., Eslami, S., Gool, L.V., et al.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**(1), 98–136 (2015)