A Pedestrian Detection Method Based on YOLOv7 Model

Binglin Li[1], Qinghe Zheng[1](✉), Xinyu Tian[1], Abdussalam Elhanashi[2], and Sergio Saponara[2]

[1] School of Intelligence Engineering, Shandong Management University, Jinan 250357, China
15005414319@163.com

[2] Department of Information Engineering, University of Pisa, 56122 Pisa, Italy

Abstract. Pedestrian detection has extensive applications in computer vision, such as intelligent transportation and security surveillance. YOLOv7 is an object detection model that achieves real-time object detection by dividing the image into grids and predicting bounding boxes and classes for each grid. This study explores and optimizes pedestrian detection based on the YOLOv7 model. Firstly, a large-scale pedestrian dataset is used to train and fine-tune the model to improve detection accuracy and robustness. The YOLOv7 model demonstrates powerful pedestrian detection performance, achieving a high average precision (AP) of 0.92 and a recall rate of 0.85. Experimental results indicate that the pedestrian detection method based on YOLOv7 achieves good results in terms of accuracy and speed, and it has high practical value and application prospects.

Keywords: Object detection · Pedestrian detection · YOLOv7

1 Introduction

Object detection is a task in the field of computer vision that aims to enable computers to automatically identify and locate objects in images, providing a foundation for many applications such as intelligent surveillance, autonomous driving, and medical image analysis [1, 2]. The significance of object detection lies in enhancing the computer's understanding and classification of complex scenes, thereby improving human-computer interaction and reducing human workload.

Pedestrian detection, as an important research direction in computer vision, has received wide attention. With the rapid development of intelligent transportation and security surveillance, accurate and efficient pedestrian detection algorithms are crucial for real-time object recognition and tracking [3, 4]. However, pedestrian detection still faces a series of challenges due to the complexity of factors such as pedestrian poses, scales, and occlusions. Pedestrians may appear in various poses, leading to changes in their appearance and shape in images, making them difficult to detect and recognize. Pedestrians can have different scales and proportions, requiring the use of multi-scale detection algorithms to detect pedestrians of different sizes. Additionally, consideration needs to be given to different image resolutions for pedestrian detection in different

F. Bellotti et al. (Eds.): ApplePies 2023, LNEE 1110, pp. 304–310, 2024.
https://doi.org/10.1007/978-3-031-48121-5_43

environments. Pedestrians may be occluded by other pedestrians or objects, making them difficult to detect. In such cases [5, 6], occlusion detection algorithms [7, 8] and corresponding processing techniques [9, 10] are needed to detect pedestrians. In recent years, object detection algorithms based on deep learning have made significant breakthroughs, and the YOLO (You Only Look Once) model has attracted attention due to its real-time performance and accuracy. The YOLO model transforms the object detection task into a regression problem by dividing the image into grids and predicting each grid's targets, significantly improving detection speed. YOLOv7, as an improved version of the YOLO series, further optimizes accuracy and efficiency, making it a powerful tool for pedestrian detection.

This paper aims to conduct research on pedestrian detection based on the YOLOv7 model. We utilize a large-scale pedestrian dataset for model training and fine-tuning to improve the accuracy and robustness of pedestrian detection. Additionally, we propose some improvement methods to enhance detection performance in challenging situations encountered in pedestrian detection.

2 YOLOv7 for Pedestrian Detection

YOLOv7 is an extended version of the YOLO (You Only Look Once) model, aimed at achieving real-time object detection without sacrificing accuracy. Similar to YOLOv4, YOLOv7 is based on the Darknet neural network framework, which includes the backbone network, intermediate network, and head network. In terms of pedestrian detection, YOLOv7 demonstrates excellent results due to its real-time performance and high detection accuracy. By utilizing pre-trained weights and transfer learning, the YOLOv7 model can be fine-tuned to detect pedestrians in different scenarios.

2.1 YOLO V7 Model Structure

YOLO v7 network model as shown in Fig. 1. It is mainly includes input (Input), backbone network (Backbone), head (Head) three parts, input layer to input picture resize of 640x640 size, input into the backbone network, Backbone module consists of several CBS convolutional layers, ELAN convolutional layers and MPConv convolutional layers, of which SBC consists of convolutional layers, batch normalization layers (Batch Normalization (BN), SiLU activation function, used to extract image features at different scales. The CBS convolutional layer is an improvement on Batch Normalization. Batch normalization refers to the standardization of the data in each batch, while the CBS convolutional layer is to standardize and scale between different feature channels, so that the features between different channels are more independent and balanced, thereby improving the generalization ability and robustness of the model. ELAN convolutional layer is a convolutional layer that introduces attention mechanism, which weights the features of different channels by calculating the similarity between different channels in the feature map, so that important features can be better captured and utilized, thereby improving the accuracy and efficiency of the model. The MPConv convolutional layer is a mixed-precision convolutional layer that accelerates the training and inference process of the model by using half-precision floating-point numbers to reduce the amount of computation and storage space for convolution computation.

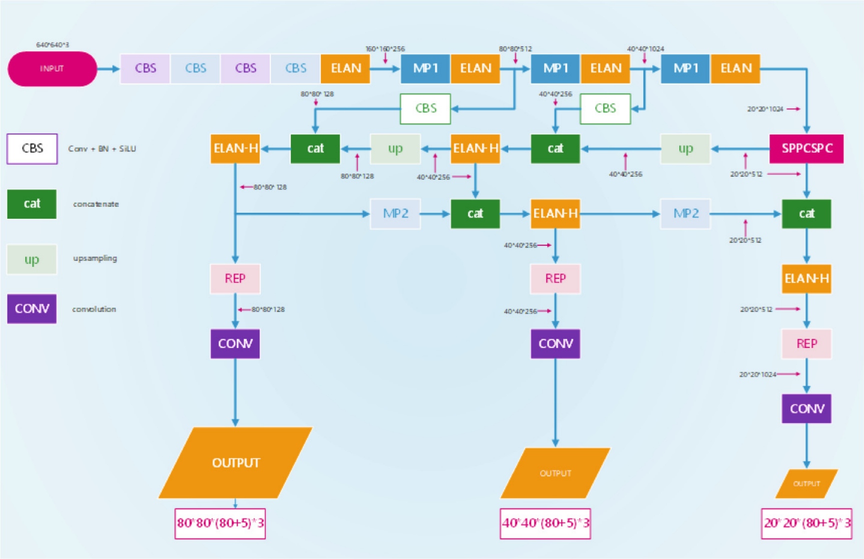


Fig. 1. Specific structure of YOLOv7.

2.2 Optimization Method

Adaptive Moment Estimation (Adam) is a commonly used optimization algorithm for training deep learning models. The specific principle of Adam algorithm is as follows:

1. For the current parameter value θ_t, calculate its corresponding gradient using batch stochastic gradient descent (mini-batch SGD), g_t.
2. Calculate first-order moment estimation: maintain an exponentially weighted moving average variable m_t, which is used to estimate the first-order moment, i.e. mean, of the current gradient. Specifically, m_t is updated as shown by

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{1}$$

3. Calculation of the second-order moment estimate: Similarly, a exponentially weighted moving average variable v_t is maintained to estimate the second-order moment (variance) of the current gradient. The update equation for v_t is given by

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{2}$$

   where β_2 is a hyperparameter ranging between 0 and 1, commonly set to 0.999.
4. Calculation of the adaptive learning rate: Based on the estimates of the first and second-order moments, the adaptive learning rate for each parameter is computed. Specifically, first, the bias-corrected values of $\widehat{m}_t$ and $\widehat{v}_t$ are calculated as

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3}$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{4}$$

Then, the adaptive learning rate α_t for each parameter is computed as

$$\alpha_t = \frac{\eta}{\sqrt{\widehat{v}_t} + \epsilon} \tag{5}$$

where η represents the initial learning rate, and ϵ is a very small constant used to avoid division by zero errors caused by extremely small denominators.

5. Parameter Update: Finally, the parameters are updated according to

$$\theta_{t+1} = \theta_t - \alpha_t \cdot \widehat{m}_t \tag{6}$$

The Adam combines momentum and adaptive learning rate to adapt to update frequencies and variations. Its basic idea is to adjust the learning rate for each parameter at each time step by estimating the first and second-order moments of the gradient.

3 Experiments and Evaluation

3.1 Experimental Setup

To validate the effectiveness of the YOLOv7 model used in this paper, we conducted experiments using a part of COCO dataset for training (2000 images for the training set, 200 for the validation set, and 200 for the test set). We selected pedestrians in different backgrounds as the research objects. The model was implemented using the PyTorch framework and trained and inferred efficiently using GPU acceleration with CUDA and cuDNN libraries. The experiments were conducted on a Windows 11 system with a batch size of 4 and 60 epochs. The experimental platform is consisted of AMD 6800HS CPU and 16GB of memory. The model was trained and predicted on an RTX3060 LAPTOP GPU with 6GB of VRAM. In the testing phase, we evaluated the trained model on a separate test set and recorded the detection results.

3.2 Evaluation Metrics

To evaluate the performance of the pedestrian detection method, we used commonly used evaluation metrics. These metrics include accuracy, recall, and average precision. Accuracy measures the overall correctness of detected pedestrian instances, while recall quantifies the method's ability to identify true positives. Average precision (AP) provides a comprehensive measure of accuracy at different recall levels, thus evaluating the model's performance comprehensively.

To evaluate the performance of our pedestrian detection method, we adopted a standard evaluation protocol. We computed the precision-recall curve, as shown in Fig. 2. Then we reported the mean average precision (mAP) at different IoU (intersection over union) thresholds in Fig. 3.

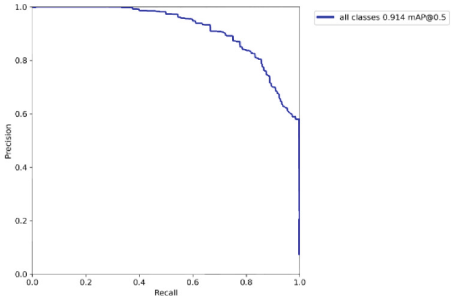


Fig. 2. Precision-recall curve.

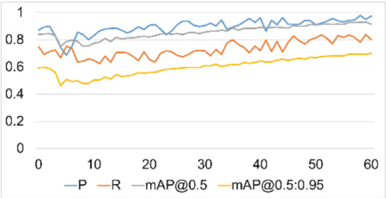


Fig. 3. The mAP at different IoU in each epoch.

3.3 Experimental Results

We present the experimental results of the YOLOv7 model on the pedestrian detection task. The model's performance is evaluated based on metrics such as accuracy, recall, and average precision (AP).

During the training phase, with only a partial COCO dataset and training for 60 iterations, the YOLOv7 model achieved an AUC of 0.914, indicating that it successfully learned to detect pedestrians from the training dataset. In the testing phase, the YOLOv7 model demonstrated strong performance in pedestrian detection. It achieved a high average precision (AP) of 0.92, indicating its ability to accurately locate pedestrians in various real-world scenarios. The model also showed a recall rate of 0.85, indicating its ability to detect most true positive pedestrian instances. The loss values of the training set (top three plots) and the test set (bottom three plots) are shown in Fig. 4. In the experiments, we evaluated the impact of different image sizes on the speed and quality of object detection. As shown in Fig. 5, larger image sizes mean the ability to detect smaller objects with higher precision, but it also increases the processing time.

The experiments are shown in Fig. 6, were conducted using images taken from four different focal lengths of the Galaxy S21 Ultra camera. Figures (a–e) show the inference results with an input size of 680 × 480, while Figures (f–j) show the results with an input size of 2560 × 1920. Figure (a) and Figure (f) were captured with a 13 mm wide-angle lens, and the model successfully detected pedestrians in the distorted regions caused by the wide-angle lens. Figures (b–c), as well as Figures (d–e), compare the results of different focal lengths of the same scene. Figure (b) has a focal length of 72 mm, Figure (c) has a focal length of 230 mm, Figure (d) has a focal length of 72 mm, and Figure (e) has a focal length of 24 mm. The model performs well under various focal lengths, even in situations with poor image quality or average lighting conditions. Comparing Figures (a–e) with Figures (f–j), it can be observed that increasing the input image size allows for better detection of pedestrians that are farther away (meaning smaller in the image), but it may result in missed detections of pedestrians closer to the camera. Overall, the method performs well under different focal lengths and various scenes.

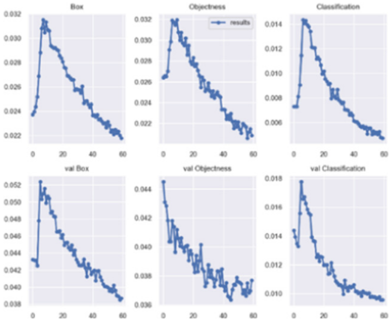


Fig. 4. The loss values of the training set and the test set.

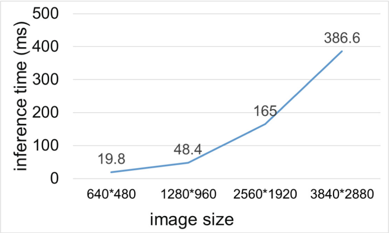
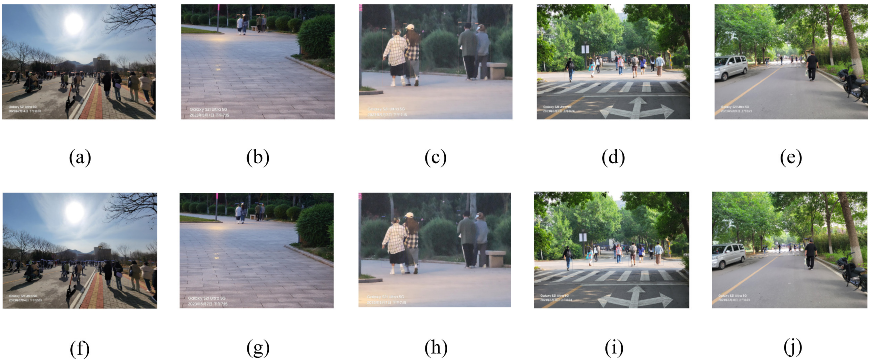


Fig. 5. The inference time of different image size.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 6. Experimental results. **a** scene 1, **b** scene 2, **c** scene 3, **d** scene 4, **e** scene 5, **f** scene 6, **g** scene 7, **h** scene 8, **i** scene 9, **j** scene 10

4 Conclusion

Pedestrian detection is an important task in the field of computer vision. Its significance lies in helping computers identify pedestrians in images or videos and perform operations such as tracking and counting. It can be applied to various practical scenarios. For example, in pedestrian safety, it can improve pedestrian safety in urban traffic. In video surveillance, pedestrian detection can help monitor densely populated areas, identify abnormal behaviors, and more. In this paper, we deployed the YOLOv7 object detection model on a computer to achieve pedestrian detection and analyzed the entire process from training to inference. In summary, our experimental results demonstrate the effectiveness and potential of the YOLOv7 model in pedestrian detection. The method achieves high accuracy and real-time performance, making it valuable in various applications such as surveillance systems and autonomous driving.

References

1. Jiao L, Zhang F, Liu F et al (2019) A survey of deep learning-based object detection. IEEE access 7:128837–128868
2. Ouyang W, Wang X (2013) Joint deep learning for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision, pp 2056–2063
3. Zheng Q, Yang M, Tian X et al (2020) A full stage data augmentation method in deep convolutional neural network for natural image classification. Discrete Dyn Nat Soc
4. Zhang Q, Yang M, Zheng Q, Zhang X (2017) Segmentation of hand gesture based on dark channel prior in projector-camera system. In: IEEE/CIC ICCC, Qingdao, China, pp 1–6
5. Li J et al (2019) Dynamic hand gesture recognition using multi-direction 3D convolutional neural networks. Eng Lett 27(3):490–500
6. Zhao L, Li S (2020) Object detection algorithm based on improved YOLOv3. Electronics 9(3):537
7. Zheng Q, Zhao P, Wang H, Elhanashi A, Saponara S (2022) Fine-grained modulation classification using multi-scale radio transformer with dual-channel representation. IEEE Commun Lett 26(6):1298–1302
8. Jiang N, Fu F, Zuo H, Zheng X, Zheng Q (2020) A municipal PM2.5 forecasting method based on random forest and WRF model. Eng Lett 28(2):312–321
9. Pathak AR, Pandey M, Rautaray S (2018) Application of deep learning for object detection. Procedia Comput Sci 132:1706–1717
10. Zheng Q, Yang M, Tian X, Wang X, Wang D (2020) Rethinking the role of activation functions in deep convolutional neural networks for image classification. Eng Lett 28(1):80–92