Accuracy and Real-Time Performance Evaluation of Pedestrian Detection Models Based on the COCO Dataset

Haoyu Wang

{whwwhy@outlook.com}

College of Computer Science and Techniques (China-Foreign Cooperation Running School), Tianjin Polytechnic University, Tianjin, China

Abstract. The detection of pedestrians holds a very important position in application fields such as self-driving cars and intelligent monitoring systems. This research carries out a comparison between a traditional method of HOG+SVM and current deep learning models, including Faster R-CNN and YOLOv8, for the purpose of assessing their detection capability. The outcome of the experiments shows that HOG+SVM tends to produce failing detections in complex environments, while deep learning models demonstrate stronger robustness. Among them, YOLOv8s attains higher detection accuracy, whereas YOLOv8n provides faster processing speed.Under the existing experimental conditions, YOLOv8 achieves a reasonable balance between precision and computational efficiency. Faster R-CNN shows relatively stable performance but lower speed due to its two-stage structure. Evaluation metrics such as mAP and FPS are used for comparison. These results provide practical guidance for selecting suitable pedestrian detection models in real-time applications.

Keywords: Pedestrian detection, target detection, Faster R-CNN, YOLOv8

1 Introduction

In recent times, the progress of deep learning has greatly promoted object detection, which at present is a basic component of computer vision. Accurate detection of pedestrians is especially very important in complex traffic situations to promote the level of safety. Nowadays, methods, which are mostly built on convolutional neural networks (CNNs), have a better effect than old feature-based ways through learning high-level expressions directly from pictures. This hereby permits dependable examination even in disordered environments or when target sizes undergo changes. The summary documents in the recent twenty years indicate that deep learning has obviously promoted the effect of examination [1]. Even so, differences still exist in correctness, calculation expenditure, and deduction velocity among various algorithms, hence comparative assessments are of value for the choosing of models.

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365342

The checking of walking persons has developed from manually made feature ways to deep learning frameworks. Early research methods united Histogram of Oriented Gradients (HOG) together with SVM classifiers, therefore giving quick detection in simple environments but having problems when occlusion, lighting changes, or complex backgrounds appear [2]. Models that are based on CNN at present hold the dominant position in this domain. Two-stage check devices, such as Faster R-CNN, first produce candidate areas, and then carry out category work and bounding box improvement work, therefore obtaining high accuracy degrees. Single-stage detection instruments, which contain YOLO modified versions, put detection as a unified regression work, therefore enabling quicker calculation output that is fit for real-time application. The recently appeared editions, such as YOLOv8, have optimized the network structure and the training method, and they have further promoted the running speed without reducing the accuracy. Besides detection, the research which are about pedestrians also includes tasks like attribute recognition, which have application uses in intelligent monitoring and city environments [3].

According to the COCO dataset, this research has built a one-class pedestrian detection experiment for the purpose of systematically making comparisons among HOG+SVM, Faster R-CNN, and YOLOv8. This analysis puts emphasis on the capability display in complicated surroundings, therefore offering reference viewpoints for actual model picking and arrangement in pedestrian check systems.

2 Data collection batch

2.1 Data set general introduction

The COCO dataset includes a great quantity of real-world pictures that have detailed marking information for every object. It is widely used in computer vision missions such as target examination and example cut division. Relevant research works have indicated that data sets that are built on the basis of the COCO format are able to offer abundant target marking information for machine learning models, and they support the research of many visual identification tasks [4, 5].

2.2 Data handling procedures

The COCO annotation document uses the JSON form, which mainly includes images, notes, and sorts. Pictures carry fundamental properties of pictures, like picture quantity, document name, and picture dimension; annotation documents preserve target annotation messages, including target category figure and corresponding bounding box location; and classifications depict objective category titles that are included in the data collection.

In the process of the experiment, the Python JSON library was utilized by us to read the annotation document and extract the annotation information which is related to the pedestrian category. Due to the fact that the YOLO family model expresses target positions in normalized coordinate values, the bounding box coordinate values that are in pixels in the COCO dataset need to be transformed into the format which is required by YOLO. Concretely speaking, the coordinate values of the bounding box must be subjected to normalization processing in accordance with the width and height of the image, therefore making their numerical range

located between 0 and 1. By this method, pictures of different distinguishing rates can keep a consistent data expression in the training process, hence enhancing the stability of the model training.

3 Investigation approaches

3.1 HOG plus SVM

The Histogram of Oriented Gradients (HOG) expresses object features through the establishment of local gradient direction distribution models. It is often used together with a support vector machine (SVM) classifier, and people widely use this combination in object detection and recognition. For instance, Document [6] utilizes HOG characteristics and SVM to finish hand area detection within the research of Gesture Recognition, and it combines tracking models and deep learning models to achieve real-time Gesture Recognition; Literature [7] utilizes HOG and linear SVM for the vehicle detection work, and thus carries out analysis on color space choice and parameter disposition. The above research indicates that the HOG+SVM method possesses certain versatility and stability in the domain of target detection.

In order to make the image ready for subsequent analysis, firstly, it is transformed into grayscale, then gradient information is got on both the horizontal axis and vertical axis:

$$G_x = I(x+1, y) - I(x-1, y) \tag{1}$$

$$G_y = I(x, y+1) - I(x, y-1) \tag{2}$$

According to the gradient components that have been calculated, the gradient's magnitude and direction can thus be further obtained in the way below:

$$G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$\theta = arcta\,n\left(\frac{G_y}{G_x}\right) \tag{4}$$

For the purpose of extracting features, the image is divided into a number of cells, in each of which the histograms of gradient directions are got. These cells are further put together into blocks, and the feature vectors undergo normalization in the corresponding way:

$$v' = \frac{v}{\sqrt{\| v \|_2^2 + \varepsilon^2}} \tag{5}$$

Finally, the feature vectors of all blocks are put together to make a whole HOG feature vector, which is fed into a Support Vector Machine (SVM) classifier to do judgment, therefore finishing the work of pedestrian target detection. Therefore, the picture gradient structure information can be utilized to carry out an efficient representation of the target shape. The flowchart of pedestrian detection, which uses HOG plus SVM, is given in Figure 1.

Fig. 1. Flow diagram of pedestrian checking through HOG+SVM (Picture credit: Original)

3.2 Faster R-CNN

Faster R-CNN is broadly utilized as a two-stage detection frame which is constituted by a backbone network, a region proposal network (RPN), and a detection module that is based on RoI. It has been employed by human individuals in numerous domains, which include medical picture analysis. As an example, it utilizes Faster R-CNN and Mask R-CNN for the detection of COVID-19 lesions in CT images, thus displaying efficacious performance [8]. At the same time, it has promoted a Faster R-CNN-dependent model for pneumonia examination through combining FPN and Soft-NMS to promote the precision of position finding [9]. On the whole, these research works thus indicate that methods that are based on region proposals possess effectiveness for complicated detection tasks.

In the stage of feature extraction, the convolutional neural network first carries out multilayer feature coding for the input image, and its convolution operation can be expressed as

$$F(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{6}$$

The input feature graph is depicted through convolution kernel parameters, hence the response at each position in the feature graph is then got.

In the stage of Region Proposal Network (that is RPN), the candidate target regions are obtained through producing a group of pre-set anchor boxes on the feature map, and the positions of candidate boxes are furthermore adjusted through bounding box regression, just like this:

$$t_x = \frac{x - x_a}{w_a}, \quad t_y = \frac{y - y_a}{h_a} \tag{7}$$

$$t_w = \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{h_a}\right) \tag{8}$$

In the training procedure, a multi-task loss function is usually utilized to carry out joint optimization of classification and regression:

$$L = L_{\text{cls}} + \lambda L_{\text{reg}} \tag{9}$$

In this place, the classification branch adopts cross-entropy loss, and the regression branch carries out regularization processing by the Smooth L1 function:

$$\text{SmoothL1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & |x| \geq 1 \end{cases} \tag{10}$$

In the last step of detection work, the candidate areas are projected onto fixed-dimensional feature expressions by the utilization of RoI Pooling or RoI Align. The entire connected layers below make the bounding boxes become more precise and forecast object categories, hence generating the final detection outcomes. Figure 2 illustrates the entire working flow of Faster R-CNN.

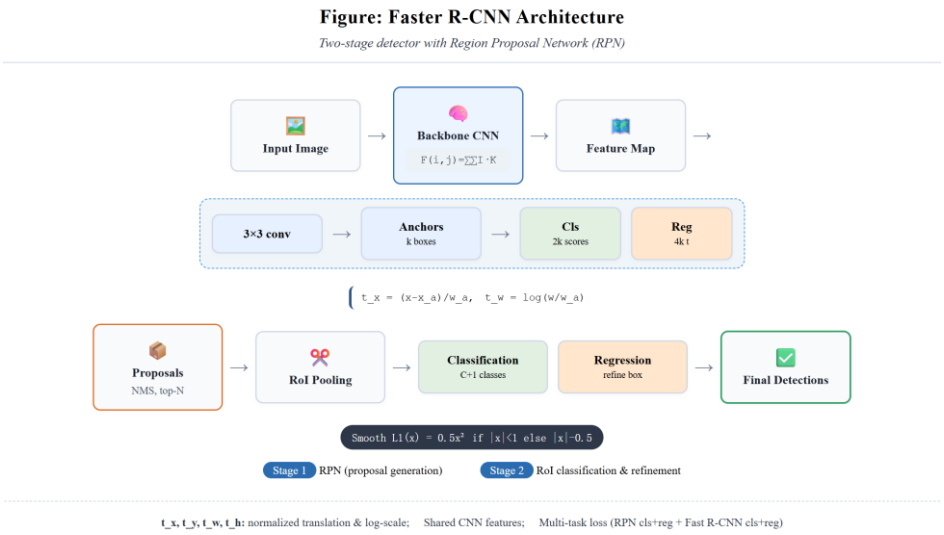

Fig. 2. The Flow Chart of Faster R-CNN (Picture credit: Original)

3.3 YOLOv8

YOLOv8 is a one-stage object detection instrument that is constituted of three main constituent parts: Backbone, Neck, and Head. The Backbone obtains level-by-level characteristics from input pictures via convolution layers, which catches meaning information on many different layers. The Neck part fuses features of multiple scales; it usually uses FPN and PAN structures, thus it strengthens the model's capability in processing objects that have different sizes. The Head uses a separated design, which processes classification and bounding box regression respectively, hence this increases detection stability [10].

The YOLOv8 algorithm has been applied by people in many different sorts of tasks. As an example, DC-YOLOv8 has carried out modifications on downsampling and feature fusion for enhancing the detection of small objects, whereas it utilizes YOLOv8 to conduct fruit maturity identification, hence attaining both high accuracy and speedy inference computation [11].

This model utilizes an anchor-not-using method, which directly carries out regression computation of object center positions and bounding box dimension sizes above the feature graph. This can decrease the dependence on pre-established anchor points and hence promote the inference efficiency. The training work usually depends on one kind of multitask loss function:

$$L = L_{\text{cls}} + L_{\text{box}} \tag{11}$$

The classification branch is carried out under supervision by means of binary cross-entropy:

$$L_{\text{cls}} = -[y \log(p) + (1 - y) \log(1 - p)] \tag{12}$$

The bounding box regression method usually uses the CIoU loss function:

$$L_{\text{box}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^*)}{c^2} + \alpha v \tag{13}$$

With regard to parameter update work, YOLOv8 in general uses the AdamW optimizer, which can be expressed as:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon} - \eta \lambda \theta_t \tag{14}$$

Where m_t and v_t are the first-order and second-order moment estimation values of the gradients, respectively. Through the combination of this network structure and the said training strategy, YOLOv8 has obtained a usable balance between detection velocity and precision, it supports effective real-time working performance. The whole working flow is shown in Figure 3.

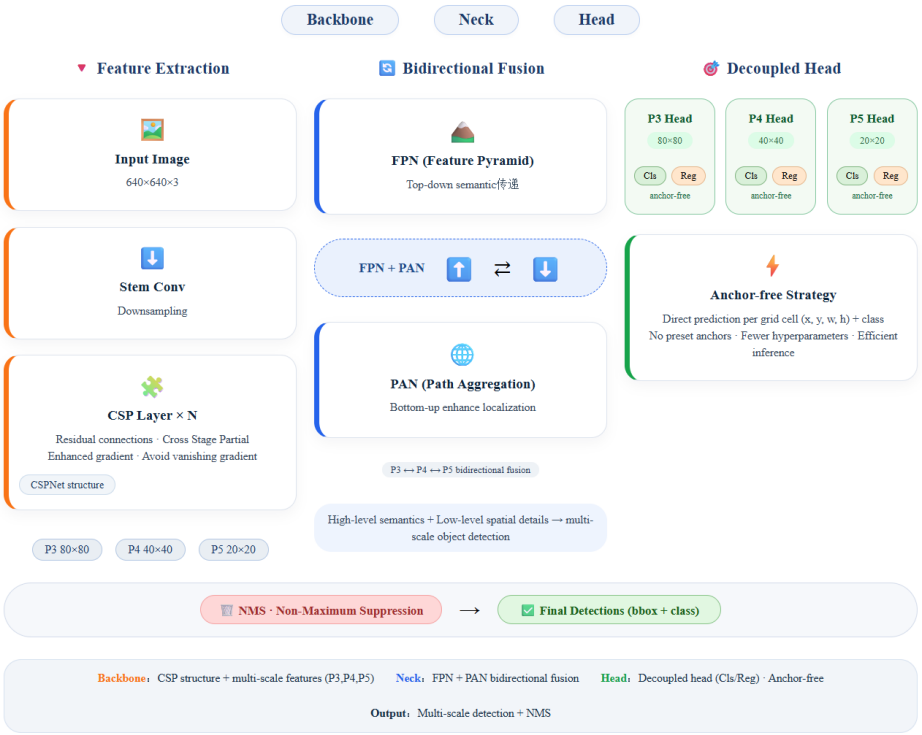

Fig. 3. The flow diagram of YOLOv8 (Picture credit: Original)

4 Experiments

4.1 Experiment surroundings and setting up

The circumstances of software and hardware. This paper carried out the experiments on a Windows 11 system and used Anaconda for the creation of a Python virtual environment that is for item management. Deep learning models have been put into practice by making use of PyTorch and Torchvision, with CUDA, which enables GPU acceleration. The work of image processing and data preparation has depended on OpenCV and Pillow, and utilized them to analyze COCO annotations and compute detection evaluation target indexes. This hardware platform was constituted by one Intel Core i7-13700H CPU and one NVIDIA RTX 4060 laptop GPU, which holds 8 GB of video memory. This arrangement offers enough calculation resources to support the training and the inference of the target detection models, while it ensures the stable running of the whole process of the experiments.

Experiment establishment.

(1) Faster R-CNN

This paper has carried out experiments on the RTX 4060 Laptop GPU (8 GB), which provides limited computation ability when put beside server-level GPUs. In order to accommodate this

situation, one subset, including 2000 images, was utilized to carry out training across five training cycles. This paper has set the batch size as 16, therefore can make the memory use and training stable to get a balance. This paper employed SGD to act as the optimizer, and it had a learning rate which was 0. 005, guaranteeing gentle convergence and preventing big undulations in the process of parameter renewal. These environmental settings enabled effective training while they checked the basic pedestrian detection ability of Faster R-CNN.

(2) YOLOv8

For the assessment of the influence that model scale exerts on detection performance, YOLOv8n and YOLOv8s have been trained and evaluated by the utilization of the same dataset splits and official pre-trained weights. The input images are processed to be resized into 640×640, whose batch size of 16, and the training is carried out over 50 epochs on the same GPU. This paper carried out performance evaluation with the utilization of COCO metrics (mAP@[0. 5:0. 95]), Exactness, Recall degree, and average reasoning time speed (FPS) on every single image. This placement lets a clear comparison of detection accuracy and processing speed between different YOLOv8 scales, hence giving guidance for real model selection.

4.2 The experiment outcomes and discussion

Table 1. The All-around Contrast of HOG + SVM, Faster R-CNN and YOLOv8

index	HOG + SVM	More Rapid R-CNN	YOLOv8n	YOLOv8s
sort of arithmetic method	traditional conventional machine study	Depth study (two stages)	Depth Study (One Stage)	Depth Study (One Phase)
characteristic extraction method	HOG manual feature	Depth feature of CNN	CNN depth characteristic	CNN depth characteristic
Precision	-	-	0.770	0.802
Recall	-	-	0.576	0.666
mAP@0.5	-	0.064	0.653	0.753
mAP@[0. 5:0. 95]	-	0.042	0.416	0.512
parameter amount	extremely diminutive	larger	3.0M	11.1M
deducing speed	27.38 FPS	14.85 FPS	417 FPS	270 FPS
Traits of Test	It is simple to realize but has not high accuracy	More stable accuracy is possessed, but the working speed is slower	Speedy and low weight	High accuracy, balanced working capability
situations that can be applied	Easy scene examination	high-accuracy work	Real-time check/embed type	High accuracy demands for real-time inspection

According to the experimental outcomes (Table 1), and can see that detection models have very big differences in both accuracy and calculation efficiency. The conventional

HOG+SVM method, which has a simple structure, obtains an inference velocity of about 27. This research got 38 FPS. However, its dependence on manually-made features makes it easy to have missed detections or wrong positive results when backgrounds are complex or when pedestrians are partially blocked, which thus limits its overall performance.

As for the Faster R-CNN, this model has obtained a mAP@[0.5: 0.95] of 0.064, which possesses an inference speed of about 14. 85 frames per second. This effect degree is lower than what was obtained from some past researches, it is mainly because the training data set is not big enough (2000 images) and the number of training epochs is few (5), therefore, these things prevent the model from achieving complete convergence. The limited hardware resources also have restricted large-scale training, hence further influencing detection outcomes.

By way of comparison, the YOLOv8 models display better performance under this experiment condition. YOLOv8s obtains a mean Average Precision@[0.5: 0.95] of 0.512, while YOLOv8n achieves a little bit lower accuracy, but it gets an advantage from a much quicker inference speed, which is about 417 FPS. This comes from its lightweight architecture. On the whole, the YOLOv8 series offers a balanced compromise between detection correctness and processing efficiency, hence it is very suitable for real-time pedestrian detection application scenarios.

4. 3 The showing of results

The research results give the implication that the method of HOG plus SVM has obtained a comparatively high reasoning speed, which is 27. Under the present experiment situation, this result gets 38 FPS, therefore it shows that it has the ability which can do real-time processing.

Fig. 4. The comparison of train outcome (JSON note on the left, model forecast result on the right)(Picture credit: Original)

From the comparison which is made of detection results in Figure 4, and can very intuitively see the performance differences of different models which are in real scenes. YOLOv 8n and

YOLOv 8s can stably give out detection frames on most test pictures, especially on complex occasions like weak light at night, many people overlapped and small goal objects. In the "crowd assemble" scene that is at the upper left corner, it is in the 268378. png picture, two editions all succeeded in recognizing many walkers, although the partial frames of YOLOv8n are a little not tight, and some frames do not correctly enclose the human body part, but the whole coverage is entire; while the bounding box of YOLOv8s is more dense and possesses higher confidence degree, hence it indicates that YOLOv8s has obtained a better balance between accuracy and speed.

5 Conclusion

This research carries out a comparison between traditional pedestrian detection methods and those that are based on deep learning, under identical experiment conditions, with the use of the COCO dataset. This experiment has proven that although HOG+SVM can give comparatively quick inference speed, its ability to detect is restricted when it is in complicated situations. Faster R-CNN shows a steady working effect, but it is influenced by not enough training data and calculation limits in this research. By contrast, the YOLOv8 series can obtain a better balance result between accuracy and efficiency, it is more appropriate to be used in real-time applications.

However, must point out that the effect of these models can have differences under different data sizes and hardware situations. More in-depth research is needed to look into bigger data collections and longer training procedures to promote model robustness and generalization capability.

References

[1] Zou, Z., Chen, K., Shi, Z., et al.: Object detection in 20 years: A survey. Proceedings of the IEEE. 111(3), pp. 257-276 (2023)

[2] Gawande, U., Hajari, K., Golhar, Y.: Pedestrian detection and tracking in video surveillance system: issues, comprehensive review, and challenges. Recent Trends in Computational Intelligence. pp. 1-24 (2020)

[3] Wang, X., Zheng, S., Yang, R., et al.: Pedestrian attribute recognition: A survey. Pattern Recognition. 121, pp. 108220 (2022)

[4] Rostianingsih, S., Setiawan, A., Halim, C.I.: COCO (creating common object in context) dataset for chemistry apparatus. Procedia Computer Science. 171, pp. 2445-2452 (2020)

[5] Le, T., Lal, V., Howard, P.: Coco-counterfactuals: Automatically constructed counterfactual examples for image-text pairs. Advances in Neural Information Processing Systems. 36, pp. 71195-71221 (2023)

[6] Huu, P.N., Phung Ngoc, T.: Hand gesture recognition algorithm using SVM and HOG model for control of robotic system. Journal of Robotics. 2021(1), pp. 3986497 (2021)

[7] Tomikj, N., Kulakov, A.: Vehicle detection with HOG and linear SVM. Journal of Emerging Computer Technologies. 1(1), pp. 6-9 (2021)

[8] Sahin, M.E., Ulutas, H., Yuce, E., et al.: Detection and classification of COVID-19 by using faster R-CNN and mask R-CNN on CT images. Neural computing and applications. 35(18), pp. 13597-13611 (2023)
[9] Yao, S., Chen, Y., Tian, X., et al.: Pneumonia Detection Using an Improved Algorithm Based on Faster R-CNN. Computational and Mathematical Methods in Medicine. 2021(1), pp. 8854892 (2021)
[10] Lou, H., Duan, X., Guo, J., et al.: DC-YOLOv8: Small-size object detection algorithm based on camera sensor. Electronics. 12(10), pp. 2323 (2023)
[11] Xiao, B., Nguyen, M., Yan, W.Q.: Fruit ripeness identification using YOLOv8 model. Multimedia Tools and Applications. 83(9), pp. 28039-28056 (2024)