Design and Implementation of Pedestrian Detection System

Hengfeng Fu[1], Zhaoyue Zhang[2], Yongfei Zhang[1], and Yun Lin[1](✉)

[1] College of Information and Communication Engineering,
Harbin Engineering University, Harbin 150001, China
linyun_phd@hrbeu.edu.cn
[2] College of Air Traffic Management,
Civil Aviation University of China, Tianjin 300300, China

Abstract. With the popularization of self-driving cars and the rapid development of intelligent transportation, pedestrian detection shows more and more extensive application scenarios in daily life, which have higher and higher application values. It also raises more and more interest from academic community. Pedestrian detection is fundamental in many human-oriented tasks, including trajectory tracking of people, recognition of pedestrian gait, and autopilot recognition of pedestrians to take appropriate response measures. In this context, this paper studies the design and implementation of a pedestrian detection system. The pedestrian detection system of this article is mainly composed of two parts. The first part is a pedestrian detector based on deep learning, and the second part is a graphical interface that interacts with the user. The former part mainly uses the Faster R-RCNN learning model, which can use convolutional neural networks to learn features from the data and extract the features of the image. It can also search the image through RPN network for areas where the target is located and then classify them. In this paper, a complete pedestrian detection system is implemented on the basis of deep learning framework Caffe. Experiments show that the system has high recognition rate and fast recognition speed in real world.

Keywords: Pedestrian detection · Faster RCNN · Target detection · Deep learning

1 Introduction

In recent years, pedestrian detection has gradually become a research hotspot due to its wide application scenarios and its fundamental role in computer vision tasks. Pedestrians are ubiquitous in our images and videos, such as photos, entertainment videos, sports broadcasts, and video surveillance and authentication systems. Because of this, pedestrian recognition is the key step when automatically understanding media content.

The usage of pedestrian detection have great application value, and can play an auxiliary role in many human-oriented tasks, such as analysis of population flow, trajectory tracking of characters, and pedestrian gait recognition. In addition, the pedestrian detection system is also an important part of the safety system of autonomous vehicles. Therefore, the pedestrian detection system is also an integral part of

S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 89–98, 2019.
https://doi.org/10.1007/978-3-030-19086-6_10

many other human-related studies, and the improvement in pedestrian detection performance may greatly contribute to the improvement of other system performance. Although pedestrian testing has been extensively studied so far, recent research can still achieve significant improvements on the basis of the previous ones, indicating that the bottleneck of pedestrian detection has not been achieved, and the pedestrian detection system still has room for improvement.

In the process of recognizing pedestrians, the variety of postures is one of the main challenges of human recognition and other object recognition. The features of pedestrians that need to be classified will change with the change of posture. The appearance of the same object will change dramatically with different postures and perspectives, which is a big challenge to the identification.

There are many ways to solve pedestrian problems. These approaches often come down to the branch of machine learning, which uses machine learning to solve pedestrian detection problems. There are many ways to use manually selected features, such as HOG features [1]. But there are also low-level features that are randomly generated, such as the Haar feature [2]. These features are applied when training to determine whether it is a pedestrian's Feller model. There are also methods such as Adaboost [3] that combine to form a strong classifier by combining multiple simple classifiers. In addition to the above methods, an important processing method is to let the classifier learn the features. For example, CNN [4] used in this paper belongs to this classifier.

These factors make the design and implementation of pedestrian detection systems important. Finding the right method to construct a pedestrian detection system is the main content of this paper.

2 Related Works

At present, pedestrian testing has achieved a lot of research results. In 2012, Dollar et al. [5] reviewed pedestrian detection and compared the best pedestrian detection methods in recent years. In 2014, Benenson et al. [6] in the field of pedestrian detection, more than 40 methods were compared on the Caltech dataset; in 2015, Hosang et al. [7] studied the application of convolutional neural networks to pedestrian detection. In 2016, Zhang et al. [8] analyses the state-of-the-arts methods and address the localization errors and background/foreground discrimination.

HOG was presented by Dalal and Triggs at the 2005 CVPR meeting. HOG stands for Histogram of Oriented Gradient. In essence, HOG is a "feature descriptor." The feature descriptor has the following features. When the same type of object is observed under different conditions and different environments, for example, objects belonging to the pedestrian category, the obtained feature descriptors are also nearly the same. The HOG feature has several advantages over the previous feature for pedestrian detection: First, since the gradient direction histogram is operated on the local square cell of the image, it makes both the geometric and optical deformation of the image. And it can maintain good invariance, because these two deformations will only appear in the larger space of the same space; secondly, under the conditions of coarse spatial sampling, fine direction sampling and strong local optical normalization, As long as the pedestrian is generally able to maintain an upright posture, the pedestrian can be allowed to have some subtle body movements, and these subtle movements can be ignored without affecting the detection effect. Therefore, the gradient direction histogram is particularly suitable for human detection in images. Although a large number of pedestrian detection algorithms have been generated from the introduction of gradient histograms to the present, many algorithms have been improved based on gradient direction histograms.

At present, the mainstream deep-based learning target detection algorithms fall into two categories: one is the region-based target detection algorithm based on Faster RCNN [9], which generates candidate target regions, and classifies the regions for detection, such as: Faster RCNN, R-FCN [10] and so on. The advantage of this type of algorithm is that the detection accuracy is high, and the disadvantage is that the speed is slow. The other type is to convert the target detection into a regression problem solution represented by YOLO (You only look once), and input the original image to directly output the position and category of the object, such as: YOLO [11], SSD [12] and so on. The advantage of this type of method is that the detection speed is fast, and the detection of several tens of frames per second can be achieved, but the detection accuracy is low, and the detection for small targets is not sensitive.

This paper introduces the Faster R-CNN general target detection algorithm into the complex scene of pedestrian detection.

3 Pedestrian Detection System Design

3.1 Regional Proposal

In the Faster RCNN model, the regional proposal is completed by the RPN Regional Proposal Network. Specifically, RPN has two kinds of prediction tasks, namely binary classification and bounding box regression adjustment. In order to train in the practical application of the pedestrian detection system, the proposed anchor frame must be divided into two different categories. The first category is the anchor box with the bounding box IoU > 0.5 marked with a target, which can be seen as “foreground”. Conversely, anchor frames that do not overlap with a bounding box or have an IoU value of less than 0.1 can be considered a “background.” In order to maintain the normal foreground and background anchor frame ratio, the system samples the randomly generated anchor frame. Then, to measure the performance of the regional proposal, the RPN network will use the binary cross entropy to calculate the classification loss.

To avoid calculating meaningless backgrounds into regression losses, the system selects those anchor boxes that are marked as foreground. To calculate the regression loss. To calculate the target of the regression, we use the candidate region labeled as the foreground, and calculate the offset between the anchor box and the bounding box of the label compared to the bounding box of the label. The error L1 uses a smooth L1 loss function. The loss function is as follows:

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & 3 \\ |x| - 0.5 & 2 \end{cases} \tag{1}$$

It's because the smooth L1 loss function decreases rapidly when the anchor frame and the bounding box are close, whereas the normal L1 loss function or the L2 loss function does not.

Due to many reasons such as the difference in image size and anchor frame size or aspect ratio, the number and position of anchor frames generated by RPN vary greatly during training. This has led to sometimes a relatively large change in the relative scale between the anchor frame of the background and the anchor frame considered to be foreground during training, and it is difficult to balance. In extreme cases, you may not even get any anchor frames that are considered foreground. In order to be able to learn at all times, a compromise method is adopted. This method sorts the anchor frames generated by the RPN according to the degree of coincidence with the bounding box of the label, and selects the box with the highest degree of coincidence as the foreground anchor box. Loss calculation. But this is not always possible.

3.2 IoU

The overlap ratio is defined by IoU (intersection over union). This indicator is used to calculate the coincidence of the border area of the label with the actual border area. It can be used to evaluate whether the bounding box matches the expectation. On a specific image, it can be used to measure the accuracy of the upper detection system. The calculation formula is as shown in the Eq. (2)

$$\mathrm{IoU} = \frac{S_I}{S_U} \tag{2}$$

The numerator is the area of the overlapping part in the bounding box of the two sides, and the denominator does not occupy the area shared by the two bounding boxes. From the perspective of the set theory, the former is the intersection of the set of pixels in the corresponding border, and the latter is the union. This is also the origin of the IoU name.

Predicting class labels are common in classic machine learning tasks, especially in categorical tasks. The result of the entire model operation is to output a two-category label, or a correct label that represents yes or no. The accuracy of the calculation of this type of two-category label is easy to understand. There are only two possibilities. When it comes to object detection tasks, the output is not that simple. In a real-world environment, the coordinates of the bounding box predicted by the detector may almost exactly match the pre-labeled coordinates representing the correct bounding box. The parameters of the detector model (image pyramid scale, sliding window size, feature extraction method, etc.) are different, and the perfect match between the predicted bounding box and the real bounding box is completely impractical. Because of this, to measure the accuracy of the bounding box, the definition of an evaluation metric is needed that rewards the bounding bounds of the predictions that overlap much with the bounding box of the label. This indicator is such that the predicted bounding box that is highly overlapping the labeled bounding box scores higher than the overlapping bounding bounding box, not the other way around. This is why IoU is an excellent indicator of the performance of a target detector. In summary, in pedestrian detection, the coordinates of the detector output are not required to match exactly with the label box, but should ensure that the bounding box predicted by the detector matches as much as possible.

3.3 Non-maximum Suppression

There are many detectors that have a three-step inspection process. The first is to search in the possible target area space based on the input image to generate a set of interest areas. The second step is to score the region of interest using a classifier or a regression. And the final step merges windows that may belong to the same object. This last step is often referred to as "non-maximum suppression". A common implementation of NMS is a simple selection of candidate boxes that are larger than a certain predefined IoU threshold, and because they may cover the same object, the candidate blocks with lower confidence levels are reduced. This algorithm is fast and simple and has significant advantages in removing redundancy.

Although the NMS process does not seem complicated, choosing a suitable IoU threshold is not straightforward for a variety of complex scenarios. When this threshold is too low, it is easy to erroneously discard many of the offer areas. The threshold of the key is set too high, and in the detection result, it is easy to obtain an offer that is too much for the same target detection area. The empirical value usually used is 0.6.

In some scenes where pedestrians are highly concentrated, such as crowded subways, mall exits, movie theater exits, etc., the threshold should be appropriately reduced. However, in the case where the target is too close, the NMS will definitely reduce the accuracy of the pedestrian detection system.

3.4 Training

It's a non-convex problem to optimizing the weight of neural networks. Therefore, in actual training, random gradient descent (SGD) or similar derivative methods such as momentum method and Nesterov's acceleration gradient (NAG) are generally used. In the gradient descent method, it is necessary to determine the learning parameters such as learning rate, weight decay and momentum. The learning rate determines the speed of weight update, and the high learning rate makes it easy for the search program to skip the optimal value. Too small will slow down the speed. Some adaptive methods are used because adjusting parameters manually requires constant modification of the learning rate.

Momentum is derived from Newton's law. The basic idea is to find the optimal "inertia" effect. When there is a flat region in the error surface, SGD can learn faster. For a certain weight, there is a formula like Eq. (3):

$$\omega_i \leftarrow \gamma \cdot \omega_i - \eta \frac{\partial E}{\partial \omega_i} \tag{3}$$

If the direction of last momentum is the same as the negative gradient direction of this time, then the magnitude of this decline will increase, so this can achieve the process of accelerating convergence.

Weight attenuation means that in practice, in order to avoid over-fitting of the network, some regular terms must be added to the cost function. For example, adding this regular term in SGD normalizes this cost function, and has a formula for a certain weight. As shown in Eq. (4):

$$\omega_i \leftarrow \omega_i - \eta \frac{\partial E}{\partial \omega_i} - \eta \lambda \omega_i \tag{4}$$

The purpose of this formula is to reduce the impact of unimportant parameters on the final result, and the useful weights in the network will not be affected by the weight decay. In machine learning or pattern recognition, over-fitting occurs, and when the network gradually over-fitting, the weight of the network gradually becomes larger. Therefore, in order to avoid over-fitting, a penalty term is added to the error function. The penalty term is the square of the weight of the property multiplied by the sum of the decay constants. It is used to punish large weights.

To perform training, we must determine the parameters for the solver of Caffe, Eq. (5) shows the weight update formula in this paper.

$$\begin{cases} v_{i+1} = 0.9 v_i - 0.0005 \epsilon \omega_i - \epsilon E\left[\dfrac{\partial L}{\partial \omega}\Big|_{\omega_i}\right] \\ \omega_{i+1} = \omega_i + v_{i+1} \end{cases} \tag{5}$$

In the network implementation part, we chose the mainstream deep learning framework Caffe as an experimental platform. According to the current standard strategy based on the deep learning target detection method, the pre-trained model initialization training network is selected on the ImageNet classification task. The ZFNet convolutional neural network pre-trained by ImageNet classification is used to initialize the weight of the feature extraction network convolution layer. The entire network training process uses SGD back propagation to optimize the entire network model. The learning rate is 0.01, the momentum is 0.9, the weight decay is 0.0005, the learning rate is attenuated every 50,000 iterations, the attenuation factor is 0.1, and a total of 100,000 iterations are performed.

We train the network in this paper with data set comes from VOC2007, the entire data train set contains 5001 images. And we choose 200 pictures from internet as the test set, The image size is various and the pedestrians are in various scales.

The equipment used in the experiment was Ubuntu 16.04 operation system and GeForce GTX 765M graphic card.

4 Experiments

According to the parameter setting in the previous chapter, the learning rate is 0.01, the momentum is set to 0.9, and the weight decay is set to 0.005. The trained model is tested on the test set, and the detection in each picture is counted. And Table 1 show all the statistical data such as the number of pedestrians in the test set, the number of pedestrians detected, the number of pedestrians missed. And the accuracy, miss rate, and false alarm rate are calculated. The table is shown in Table 1.

Table 1. Pedestrian detection system detection result

Total	Detected	Undetected	Error-detected	Recall	Miss rate	False alarm rate
221	218	3	19	98.63%	1.36%	8.01%

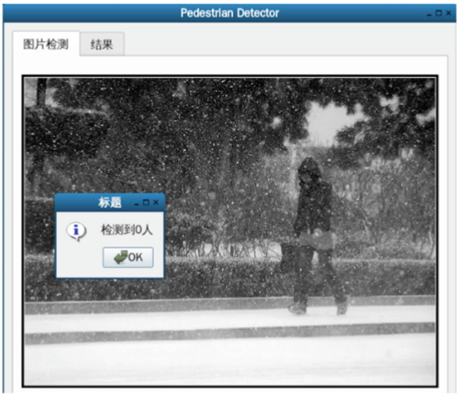
As can be seen from Table 1, there are 221 pedestrians in the test set which includes 200 pictures. The pedestrian detection system detects 218 pedestrians among them, and 3 pedestrians are not detected, and the number of pedestrians detected incorrectly is 19. The calculated results show that the detection accuracy of the pedestrian detection system is 98.63%, the detection failure rate is 1.37%, and the false alarm rate is 8.01%.

4.1 Influence of Color on Pedestrian Detection System

In the pedestrian detection system, the device that collects the images may be able to acquire RGB image, but it can only acquire grayscale images too. In order to study the effect of color on the performance of the pedestrian detection system, the sample image is first converted into a grayscale image. Then, the number of pedestrians detected, the number of pedestrians detected, the number of pedestrians missed, and the relevant parameters are calculated. The results are shown in Table 2.

Table 2. Comparison of detection result on RGB image and grayscale image

Image	Total	Detected	Undetected	Error-detected	Recall	Miss rate	False alarm rate
RGB	221	218	3	19	98.63%	1.36%	8.01%
Grayscale	221	214	7	42	96.83%	3.17%	16.41%



检测到0人

Fig. 1. Detection result of the grayscale image

From the comparison of the RGB image and the grayscale image, it can be seen that after losing the color information, the detection accuracy is reduced, and the false alarm rate is increased. The reason can be inferred from the following figures, Figs. 1 and 2:

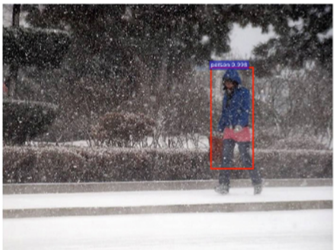


Fig. 2. Detection result of the RGB image (Color figure online)

As can be seen from Fig. 1, pedestrians walking on snowy days are grayed out because it is snowing. Therefore, the blue down jacket and pink skirt worn by pedestrians are very contrasting with the background. Comparing the Fig. 2, we can find that there is no color information in the grayscale image Fig. 1, and it can seen from the Fig. 1 that the gray color of the pedestrian lower body is similar to the gray background of the green belt, the upper body and the background of the trees, and the edge is blurred. The above reasons have led to a decline in the performance of pedestrian detection systems.

4.2 The Effect of Image Compression Quality on Pedestrian Detection System

In order to study the effect of image compression quality on the quality of pedestrian detection system, we re-compress the sample image by 80%, 60%, 40%, 20%, 10% quality parameters, and input the model for detection and statistics. The number of pedestrians present, the number of pedestrians detected, the number of pedestrians missed, and the relevant parameters were calculated. The results are shown in Table 3.

Table 3. Comparison of detection results of different compression quality downlink detection systems

JPEG quality factor	Total	Detected	Undetected	Error-detected	Recall	Miss rate	False alarm rate
100%	218	3	19	98.63%	1.38%	8.01%	8.01%
80%	218	3	19	98.63%	1.38%	8.01%	8.01%
60%	215	6	20	97.28%	2.71%	8.51%	8.51%
40%	215	6	20	97.28%	2.71%	8.51%	8.51%
20%	214	7	64	96.83%	3.17%	23.02%	23.02%
10%	210	11	142	95.02%	4.98%	40.34%	40.34%

It can be seen that although the image compression is very serious, the image detection accuracy remains high, showing the considerable stability of the detection algorithm. When the compression exceeds a certain threshold, it is 20% in this test set, and the detection false alarm rate will increase rapidly. This reminds us that the image quality should be kept at a reasonable level during pedestrian detection.

5 Conclusion

The pedestrian detection system in this paper has good stability for the pedestrians and achieve a detection rate of more than 95% and a false alarm rate and false alarm rate of less than 10%. At the same time, through experimental analysis, the system still has a good detection rate for grayscale images and low quality images, and has certain anti-interference ability, but the system still has room for improvement. Under the condition of using GPU acceleration, it takes 0.3 s–0.5 s per image for detecting in the detection process of the system. In the real world, the speed cannot meet the needs of real-time detection.

Acknowledgment. This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

Meantime, all the authors declare that there is no conflict of interests regarding the publication of this article.

We gratefully thank of very useful discussions of reviewers.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, p. I. IEEE (2001)
3. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. J. Physiol. **195**(1), 215–243 (1968)
4. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
5. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012)
6. Benenson, R., Omran, M., Hosang, J., Schiele, B.: Ten years of pedestrian detection, what have we learned? In: Agapito, L., Bronstein, Michael M., Rother, C. (eds.) ECCV 2014, Part II. LNCS, vol. 8926, pp. 613–627. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_47
7. Hosang, J., Omran, M., Benenson, R., Schiele, B.: Taking a deeper look at pedestrians. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4073–4082. IEEE (2015)

8. Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B.: How far are we from solving pedestrian detection? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1259–1267. IEEE (2016)
9. Ren, S., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
10. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: 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 (2016)
12. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part I. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2