Artificial Intelligence Technologies and Applications
C. Chen (Ed.)

doi:10.3233/FAIA231325

People Flow Monitoring Based on Deep Learning

Yunfan LOU[1]
School of Information Science and Engineering, Shandong University Qingdao, Shandong, People's Republic of China, 266237

Abstract. According to the application scenarios of the size of the human flow in different consumption places, to solve the problem of crowd detection, distance estimation between crowds, and the inability to monitor and calculate the human flow in real-time, this paper designs a real-time crowd detection scheme for the application scenarios where consumers pay attention to the size of the human flow in consumption places. The main use of the YOLO algorithm with the Darknet53 network as the main network is to separate pedestrians from the background. Pedestrians' central two-dimensional coordinates are converted into three-dimensional coordinates, realizing crowd detection and, apart from the distance estimation of crowds, real-time monitoring of current regional traffic and flow density and solving the problem of being unable to monitor and calculate people in real-time. It can be applied to many aspects, such as shop rating, traffic control, and flow control of scenic spots. Existing monitors are affected by different lights and cannot provide accurate data. In addition, the processing algorithm of this scheme is stable and accurate, and preprocessing is done before judging the human flow and the position of the human body to reduce the interference of light. This scheme has the performance of real-time monitoring and calculation through experimental verification.

Keywords. Target detection; perspective transformation; distance detection

1. Introduction

With the continuous improvement of the economy and people's level, more and more consumers are turning to cultural and spiritual consumption. However, with the increase in the flow of people, there are frequent occurrences of dense crowds in streets, scenic spots, and other places in various places, and queuing up for purchases at certain shops overnight. Especially in the season of high incidence of infectious diseases, dense crowds are likely to increase the risk of disease transmission, so a product is urgently needed to control the flow of people in related places.

Human flow monitoring methods can be divided into two categories: image-based and non-image-based. Image-based methods usually need to process the image to extract the head and shoulder features to realize the statistics of people and human flow monitoring. Object detection is one of the core problems in computer vision, and it has a wide range of applications in image, intelligent monitoring, etc. Target detection algorithms can be divided into feature extraction-based algorithms [1] and

[1] Corresponding author: Yunfan LOU, School of Information Science and Engineering, Shandong University Qingdao, Shandong, People's Republic of China, 266237, email: 202000120258@mail.sdu.edu.cn

convolutional neural network-based target detection [2]. The target detection algorithm uses feature extraction operators such as SITF [3], LBP [4], HOG [5], or Haar [6] to extract features from target candidate regions and classifiers such as SVM to detect and classify targets. Felzenszwalb et al. [7] combined HOG with SVM to propose a deformable part model DPM, which stands out among object detection algorithms. Although the algorithm has made some achievements, it has the disadvantages of high time complexity and many redundant windows, and the manual design features have low robustness, low detection accuracy, and poor generalization. With the rapid development of deep learning, by introducing deep semantic features, convolutional neural networks for target detection have shown great advantages compared to algorithms. Among them, the YOLO algorithm proposed by Redmon et al. [8] is the current mainstream target detection algorithm.

This paper mainly proposes a real-time crowd-monitoring scheme based on the YOLO algorithm. The YOLO algorithm is used to detect pedestrians and calculate the distance between pedestrians, which is of great significance for calculating the crowd gathering situation in public places and the social safety distance of crowds.

2. Key Technology

2.1 Addition of Distance Detection

In this paper, the center coordinates of pedestrians in images are detected, and the distance between pedestrians is calculated according to the distance between the center coordinates of pedestrians. The key problem distance detection faces is converting between two-dimensional pixel coordinates and three-dimensional world coordinates. In this paper, the center point of a two-dimensional image is taken as the origin to establish a coordinate scheme and a pixel value is taken as a basic unit. The distance between the bottom center coordinates of the image is used to represent the actual distance of pedestrians in the street. The relationship between the two is related to the height, position, viewing angle range, and inherent parameters of the camera. In this paper, the pinhole camera model [9] maps the two-dimensional pixel coordinates to the corresponding three-dimensional world coordinates using the direct linear transformation method. The transformation relationship between the coordinates is shown in Figure 1.

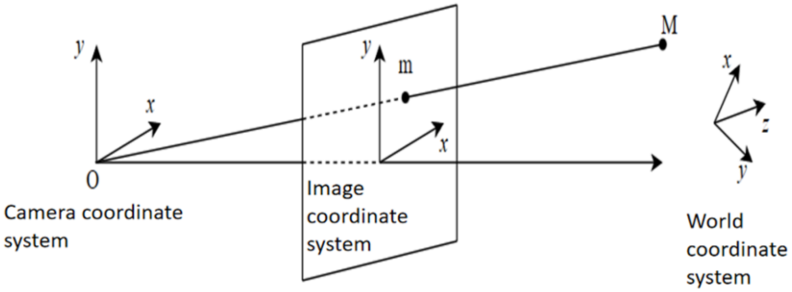


Figure 1 Transformation relationship between coordinates

2.2 Stereo Matching Algorithm Based on Deep Learning

A stereo matching network based on deep learning requires high real-time performance. The running speed of BGNet is 39 fps, which meets the requirement of real-time in automatic driving scenes. BGNet designs a cost up-sampling module based on two-sided meshes, which uses two-sided meshes and slicing operations to up-sample low-resolution cost quantities with high-resolution cost quantities. The module uses 3 × 3 × 3 3D convolution to convert the low-resolution cost feature map into bilateral mesh, where the input of 3D convolution is the cost of dimension (X, Y, D, C) and the output is the bilateral mesh feature map of dimension (X, Y, D, G). D represents the disparity map. C represents the number of channels, and G represents the guiding feature [10-12]. The high-dimensional feature map can be transformed into the leading feature G by two successive 1 × 1 convolutions. Finally, the slice operation based on linear interpolation is used for up-sampling. The cost quantity up-sampling module based on two-sided meshes greatly reduces the computation. It is easy to transplant and can be used in other stereo matching networks. The diagram is shown in Figure 2

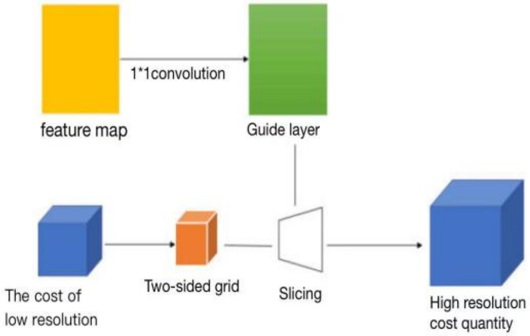


Figure 2 Cost Quantity Upsampling Module (CUBG) based on bilateral grid

3. Experiments and Results

3.1 Experiment

The experiment in this paper is Windows 10. The operating scheme experiment hardware platform processor is Intel Core i9-9900XRAM 64 GB. The graphics card model is NVDIA GeForce RTX 2080TI × 2 video memory 22 GB. The language is python3.7, and the neural network is built based on PyTorch deep learning framework.

3.2 Experiments and Results

The data set used in this paper is the public data set MOT15, which contains 22 videos, including 11 videos as training sets and 11 as verification sets. All videos are divided into frames. Images are uniformly named as 6 digits in jpeg format, such as 000001. Jpg. Some data samples in the MOT15 dataset are shown in Figure 3.

Figure 3 Sample data in MOT15

Through data enhancement and random scrambling data enhancement, including image mixing, random pixel transformation, random clipping, random horizontal flipping, and other enhancement model robustness to avoid over-fitting, we use DarkNet 53 network model as the backbone network and use YOLOv3 framework. The input image size is 512 × 512 batches. They are set to 35 iteration rounds. They are set to 1000 rounds. We select Momentum to optimize the initial learning rate. It is set to 0.00025. The learning rate is the decay rate. It is 0.1. We pre-train the model on Microsoft COCO data set [13]. The accuracy rate of the final experiment is 84.3%, and the recall rate is 87.5%. The calculation formula for the accuracy rate and recall rate is as follows:

$$\begin{cases} precision = \dfrac{TP}{TP+FP} \\ Recall = \dfrac{TP}{TP+FN} \end{cases} \tag{1}$$

TP is the number of true positives. *FP* is the number of false positives. *FN* is the number of false negatives. The target detection effect is shown in Figure 4, in which the red mark is the gathering crowd with the pixel value as the unit. And the data, such as coordinates and distance, are shown in Table 1. At the same time, Figure 4 shows the pedestrian distance detection in a two-dimensional image, that is, Euclidean distance, which is different from the pedestrian distance in the three-dimensional real world. We will explain how to convert two-dimensional coordinates into three-dimensional coordinates and calculate the distance below.

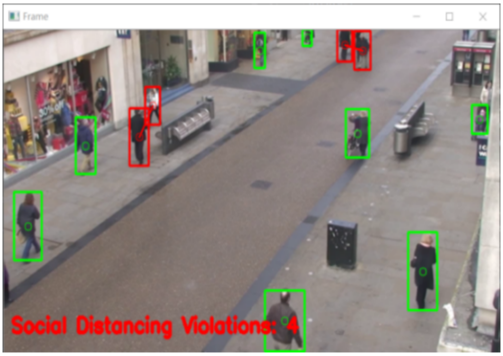


Figure 4 Target detection renderings

Table 1 Data related to gathering population

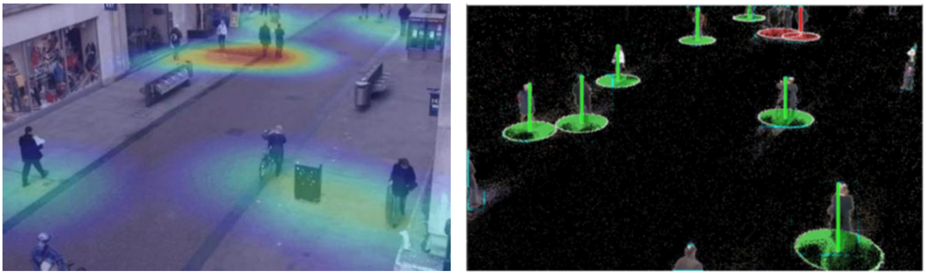
Pedestrian number	IoU	Upper left corner coordinates of the border	Coordinates of the lower right corner of the border	Border center coordinates	Distance
2	0.9908	(466, -1)	(487, 40)	(476.5, 19.5)	37.6430
9	0.8768	(425, -2)	(438, 19)	(431.5, 8.5)	
8	0.9076	(484, -1)	(505, 41)	(494.5, 20.0)	18.3371
10	0.7556	(504, 2)	(521, 45)	(512.5, 23.5)	

Because we can't know the camera height, visual angle range, and inherent parameters of the camera in the data set, it is difficult to convert the pixel distance into the actual distance, so this paper directly sets 100 basic units as 1 meter by calculating our perspective transformation matrix as:

$$\begin{bmatrix} 0.8092 & -0.2960 & 11.0 \\ 0.0131 & 0.0910 & 30.0 \\ 0.0001 & -0.0052 & 1.0 \end{bmatrix}$$

Through perspective transformation, the center three-dimensional coordinates of each pedestrian are obtained and selected to be 0 so that the center of each pedestrian detection frame is converted to the bottom center of each pedestrian, and the distance calculation of the three-dimensional coordinate scheme is converted to the bottom center of each pedestrian. At the same time, the difference in height between pedestrians is eliminated. We detect whether each pedestrian is connected with the bottom circle of others within a radius of 0.5 meters. That is 50 basic units. If so, we mark the two sides. The measurement effect is shown in Figure 5, which shows that (a) it is the heat map of the distance between pedestrians in the street, (b) it is the image background removal effect to realize the image enhancement map and (c) it is the street real scene crowd gathering detection effect map. For the detection of distance, we numbered each pedestrian and calculated the distance between pedestrians with L2 normal form, regarded each pedestrian as a class, and calculated its distance with other classes by clustering. The two classes that are too close to each other are merged into a new class. The mathematical formula of the L2 normal form is as follows:

$$\|D\|_2 = \sqrt{\Sigma_{i=1}^{3}(q_i - p_i)^2} \tag{2}$$



(a) Heat map of crowd concentration degree

(b) Background removal effect image

Figure 5 Detection renderings

4. Conclusion

This paper mainly introduces a real-time crowd detection scheme. The scheme uses the YOLO algorithm to detect pedestrians and estimates the three-dimensional real distance between pedestrians in two-dimensional images through perspective transformation technology. This scheme has achieved good results in crowd detection and crowd distance estimation through experiments. In the future, we will try new target detection algorithms and the technology of transforming two-dimensional coordinates into three-dimensional coordinates to optimize the scheme and provide more accurate services continuously.

References

[1] Jingcheng Z, F., Xinru F U, Zongkai Y, S.: UAV detection and identification in the Internet of Things [C] 1499-1503 (2019).

[2] Dong Wenxuan, F., Liang Hongtao, S., Liu Guozhu T. Review of Deep Convolution (2022).

[3] Marius M, Paul B, Ciprian M. Weigh-in-Motion Sensors and Traffic Monitoring Systems. State of the Art and Perspectives (2022)

[4] Wang Songtao, F., Gan Xudong, S., Wang Li. T. Traffic monitoring system design and implementation based on LabVIEW (2022).

[5] Bahaa A A, Takeshi T. Traffic Monitoring System Based on Deep Learning and Seismometer Data. (2021)

[6] Jr D P E, Hara C, Jr T P, et al. BackStreamDB: A stream processing engine for backbone traffic monitoring with anomaly detection. (2020)

[7] King. F. Traffic monitoring method based on the CSI research (2021).

[8] Li Ao, F., Zong Feng, S.: Traffic monitoring system based on deep learning study (2021).

[9] Wang Chongguo, F., Shi Gang, S., Chen Tianxi, T. Traffic monitoring system based on OpenCV (2021).

[10] Wang Cune, F., Yang Yanning, S., Ren Xincheng, T. Indoor traffic monitoring system design (2018).

[11] Shi Yanqing, F., Chang Caixia, S., Liu Xiaohong, T. Advances in calibration methods of internal and external parameters of planar array cameras (2021).

[12] Jia Shi-na. F. The small target detection algorithm based on improved YOLOv5 research (2022).

[13] Huang. F.: Target detection based on deep learning applied research. (2022).