ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

Pedestrian detection based on multi-layer feature fusion

Ruolan Deng[1, *], Biqiang Guan[2], Ziteng Li[3], Jiahao Wang[4]

[1]University of Leeds, Leeds, United Kingdom
[2]Beijing University of Chemical Technology, Beijing, China
[3]Beijing Information Science and Technology University, Beijing, China
[4]Zhejiang University, Hangzhou, Zhejiang, China
**Corresponding author: ruolandeng97@gmail.com*
These authors contributed equally to this work

Abstract: *This paper proposes a goal detection network of end-to-end multi-scale feature fusion because of the tiny pedestrian target and blocking in pedestrian detection. This algorithm is based on the YOLOv3 network, fully integrates multi-scale features, enhances the expression ability of small target features, improves the robustness of pedestrian detection in complex environments, and improves pedestrian detection accuracy based on guaranteeing real-time detection. In the experiment, the current mainstream pedestrian detection algorithm is compared. This algorithm effectively improves the detection accuracy in INRIA and KITTI data sets, and the average accuracy of Yolov3 in two different data sets is improved by 6% and 24.7%, respectively.*

Keywords: *Driverless, Pedestrian Detection, Target Detection, YOLOv3*

1. Introduction

Pedestrian detection, as one of the essential tasks of unmanned driving systems, monitoring systems, and early warning protection systems, plays an essential role in many fields. With the development of science and technology in China, high-speed EMUs and new-type automobiles have gradually emerged on the roads, alongside the increasing population density and the total number of vehicles. However, there are always many drivers who do not obey the rules. The severe traffic situation inevitably poses a significant threat to the personal safety of pedestrians [1]–[4]. At present, in driving a car, the driver judges whether a pedestrian is ahead to take emergency measures. However, the pedestrian target is often tiny and blocked, so when the road is curved. The driver's sight is more easily affected, it is difficult to ensure the safety of pedestrians. Therefore, it is essential to develop a theory and method that can accurately and quickly detect pedestrians on the road, as it can ensure the safety of automobiles and protect the life and property of the public, which is of great practical significance.

In response to the above issue, based on the improved YOLOv3 network, we came up with an end-to-end target detection network for pedestrian detection for in-vehicle scenarios. Incorporating multi-scale features, the network enhanced the expression ability of small target features, the robustness of pedestrian detection in complex environments, and pedestrian detection accuracy based on real-time detection.

The main contributions of this paper are as follows:

(1) We improved YOLOv3 and proposed an end-to-end target detection network incorporating multi-scale features. In the meantime, we encoded the semantic dependence between space and channels and incorporated shallower spatial features with deep semantic features. This helped us access a significantly improved detection accuracy rate of small targets without basically increasing the amount of calculation.

(2) On a large-scale data set covering various environments and pedestrian types, we validated our improved network. We used the mosaic data enhancement method to further enrich the training samples without increasing the training time. On the detection side, K-means clustering was used to obtain anchor box parameters that are easier to learn and collect to receive a more accurate prediction of the location of the target area in the subsequent regression calculation.

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

2. Related works

(1) Traditional pedestrian detection method

Many scholars have done much research on pedestrian detection, which can be divided into traditional pedestrian detection methods and deep learning-based pedestrian detection methods. Moreover, early scholars primarily studied the former. The standard features commonly used in traditional algorithms are divided into Histogram of Oriented Gradient (HOG) [5], Viola-Jones [6], [7], Detection Algorithm (referred to as V-J algorithm) and DMP [8] Algorithm. Dalal and Triggs [5] proposed the Histogram of Gradients (HOG) to represent the local variance of an image and combined it with a support vector machine for pedestrian detection. Paoletti et al. made improvements based on HOG [9], using the Haar classifier to generate pedestrian candidate sequences and verify them, but could not ensure real-time performance. There was still much room for improving detection accuracy. Kartika Candra Kirana et al. improved the V-J algorithm by determining the optimal scale factor [10]. Although it could reflect the light and dark changes in a local area and improve the real-time performance of the algorithm, it still had the shortcomings of average accuracy and insufficient robustness. Zeng Jiexian et al. optimized DPM by dividing pedestrians into separate and mixed distributions for detection [11]. For the recognition of daily traffic environment, the robustness of pedestrian detection was high, but the detection effect of pedestrians still needed to be verified in low light or complex background. The method mentioned above of using traditional manual feature extraction to achieve pedestrian detection relied too much on the designer's experience and could not adaptively extract features. It is challenging to retrieve high-level semantic information in complex and changeable scenes fully, and the scene migration ability is insufficient. Therefore, it is challenging to design a new type of pedestrian recognition model to achieve efficient and accurate target detection in a complex and changeable traffic environment.

(2) Pedestrian detection algorithm based on deep learning

With the rapid development of deep learning, we can gain access to impressive detection speed and accuracy of the adaptive feature extraction algorithm based on the convolutional neural network [12] and breakthrough traditional manual feature construction limitations. The target detection algorithm can be roughly divided into two categories: a two-stage target detection algorithm and a one-stage target detection algorithm. Two-stage methods such as R-CNN [13], Fast-CNN [14], Faster R-CNN [15], Mask R-CNN [16] use region suggestions to generate candidate regions and then use CNN for target detection. Although these methods have high accuracy, their detection speed is plodding due to overly complex calculations, so they cannot achieve real-time detection targets in practical applications. As to single-stage algorithms such as YOLO [17]–[19] and SSD [20], the detection problem is treated as a regression problem, which can directly predict the location and category of the target. Compared with the two-stage one, they have a more straightforward algorithm structure and greatly improved real-time performance but with lower accuracy. HAN et al. [21] proposed a visual detection fusion system, which effectively reduced the missed detection rate by improving the YOLO algorithm. However, this method is greatly influenced by illumination and has a poor detection effect of small objects. KUANG et al. [22] increased the number of network layers of YOLOv3 and redefined the loss function. Although the detection accuracy of small target pedestrians is effectively improved, there is much room for improvement in detection speed, thus unable to realize real-time detection of pedestrians. Li Fujin et al. [23] used deep decomposable convolution as the backbone network extraction feature of SSD, which improved the real-time performance of the algorithm. However, the detection accuracy of the algorithm was too low to meet the needs of accurate pedestrian detection in actual scenes. This paper optimizes the YOLOv3 algorithm and proposes a new end-to-end real-time pedestrian detection algorithm, which can significantly improve pedestrian detection accuracy based on real-time detection and realize real-time detection of small targets pedestrians in complex environments with high precision and robustness.

3. Method

3.1 Network backbones

The YOLOv3 network is a single-stage target detection method; unlike the target detection framework of the R-CNN series, the YOLOv3 network does not generate candidate boxes and returns the location of the bounding box and its category directly at the output layer [19]. YOLOv3 draws on the idea of ResNet, the FPN network, to add cross-layer jump connections, combining the characteristics of coarse and fine granularity and better enable detection tasks [24], [25]. Add multi-scale predictions, i.e. forecasts at three different feature layers and scale predictions with three anchor boxes. The anchor box

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

is designed using clustering, divided into 9 cluster centres and divided into three feature layers equally by size. The dimensions are 13 x 13, 26 x 26, 52 x 52.

The feature extraction network of YOLOv3 is Darknet-53, and its network structure is shown in Table 1. Convolutional in the Darknet-53 network represents a CBL operation that includes volume base, batch normalized BN layer, and Leaky_Relu activation functions. For YOLOv3, the BN layer and the Leaky_Relu are inseparable parts of the volume base, forming the minor components together. In addition, the Resn Residual module is included, with the numbers 1, 2, 8, 8, 4 in Table 1 representing the number of residual units.

Darknet-53 has deepened network structure, the processing speed of 78 graphs per second, slower than Therknet-19, but one times faster than the resNet-152 with the same accuracy, so Darknet-53 is a feature extraction network architecture that takes speed and precision into account [19].

Table 1: The feature extraction network of YOLOv3

Layer	Filter size	Repeat	Output size
Image			416*416
Conv	32 3*3/1	1	416*416
Conv	64 3*3/2	1	208*208
Conv	32 1*1/1	Conv	208*208
Conv	64 3*3/1	Conv x 1	208*208
Residual		Residual	208*208
Conv	128 3*3/2	1	104*104
Conv	64 1*1/1	Conv	104*104
Conv	128 3*3/1	Conv x 2	104*104
Residual		Residual	104*104
Conv	256 3*3/2	1	52*52
Conv	128 1*1/1	Conv	52*52
Conv	256 3*3/1	Conv x 8	52*52
Residual		Residual	52*52
Conv	512 3*3/2	1	26*26
Conv	256 1*1/1	Conv	26*26
Conv	512 3*3/1	Conv x 8	26*26
Residual		Residual	26*26
Conv	1024 3*3/2	1	13*13
Conv	512 1*1/1	Conv	13*13
Conv	1024 3*3/1	Conv x 4	13*13
Residual		Residual	13*13

3.2 Multi-layer feature fusion

Feature fusion integrates different types and scales of features, removing redundant information to get better feature expression [26]. Intuitive fusion in neural networks is generally divided into Add and Concatenate. Add method is the combination of feature diagrams so that the layer frame describes the amount of information of image features; that is, the dimension of the image itself has not increased, but the amount of information under each dimension has increased, such a fusion method is conducive to the image classification task. Concatenate is a combination of channel numbers, i.e. the characteristics that describe the image itself are increased, and the information under each feature does not increase. Our algorithm uses Concatenate to fuse features.

The shallow features extracted by neural networks have high resolution to learn spatial features in an image, and the lower resolution of deep features can learn better semantic features. In order to better combine shallow images with deep images, we try to change the network structure and combine in-depth features with shallower features.

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

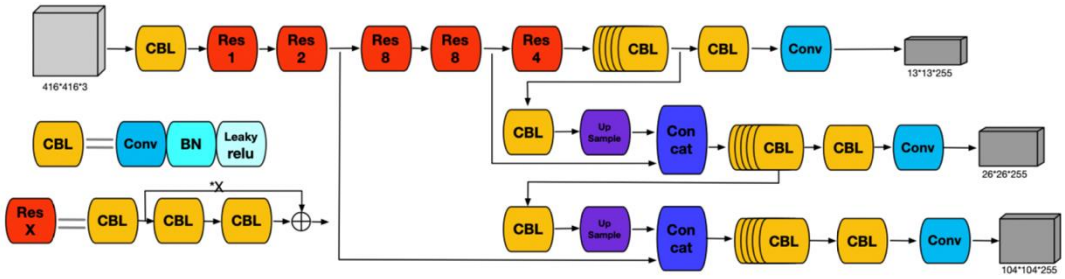

Figure 1: The network backbone of convolutional neural networks

3.3 Loss function

The loss function of the algorithm consists of three parts, which are the positioning error of the bounding box, the error of confidence of the bounding box and the classification error of the bounding box,

Positioning error of the bounding box:

The positioning error of the bounding box mainly includes the centre coordinate error and the vast and high coordinate error， which represents when the j th anchor box of i th grid is responsible for a real target, then the box generated by this anchor box should be compared with the box of the actual target, and the central coordinate error and the wide-height coordinate error are calculated.

The parameterI_{ij}^{obj} means the j th anchor box of the the i th grid is responsible for this object or not. If the parameter responsible for the object, I_{ij}^{obj}= 1. Otherwise, I_{ij}^{obj} = 0.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{obj}\left[\left(x_i-\hat{x}_i^j\right)^2+\left(y_i-\hat{y}_i^j\right)^2\right]+$$

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{obj}\left[\left(\sqrt{\omega_i^j}-\sqrt{\hat{\omega}_i^j}\right)^2+\left(\sqrt{h_i^j}-\sqrt{\hat{h}_i^j}\right)^2\right]$$

Confidence error of the bounding box:

Confidence indicates how confident the box is that there is indeed an object in the box and how confident the box is that the box includes all the characteristics of the entire object. Confidence errors are calculated regardless of whether or not the anchor box is responsible for a goal. Confidence errors are represented by cross-entropy.

In the parameter $\hat{C}_i^j$represents the actual value, and the value of $\hat{C}_i^j$ is determined by whether the value of the grid cell's booking box is responsible for predicting an object. If responsible, then $\hat{C}_i^j$ = 1, otherwise$\hat{C}_i^j$ =0.

$$\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{obj}\left[\hat{C}_i^j\log(C_i^j)+\left(1-\hat{C}_i^j\right)\log(1-C_i^j)\right]+$$

$$\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{noobj}\left[\hat{C}_i^j\log(C_i^j)+\left(1-\hat{C}_i^j\right)\right]\log(1-C_i^j)$$

It should be noted that the loss function is divided into two parts: there are objects, there are no objects, there is no object loss part also increased the weighting factor. The reason for adding a weight factor is that for an image, most of the content generally does not contain objects to be detected, which results in no objects contributing more computationally than there are objects, which causes the network to tend to predict that cells do not contain objects. Therefore, reduce the contribution weight of the calculated portion of no object, such as the value is 0.5.

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

Classification error of the bounding box:

Classification error is also chosen as the loss function of cross-entropy. When the j th anchor box of the i th grid is responsible for a real target, then the bounding box generated by the anchor box will calculate the classification loss function.

$$\sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} [\hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j)]$$

The loss function of YOLO v3 can be obtained by three parts as follows:

$$Loss = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[\left(x_i - \hat{x}_i^j \right)^2 + \left(y_i - \hat{y}_i^j \right)^2 \right] +$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[\left(\sqrt{\omega_i^j} - \sqrt{\hat{\omega}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] -$$

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[\hat{C}_i^j \log(C_i^j) + \left(1 - \hat{C}_i^j\right) \log(1 - C_i^j) \right] -$$

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{noobj} \left[\hat{C}_i^j \log(C_i^j) + \left(1 - \hat{C}_i^j\right) \right] \log(1 - C_i^j) -$$

$$\sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} [\hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j)]$$

4. Experiments

4.1 Implementation details

In this experiment, we use the PyTorch framework to implement the algorithm, and the data set is the pedestrian detection data set of INRIA and KITTI. The INRIA data set includes 614 training sets and 288 verification sets, and there are up to 15 cars and 30 pedestrians in each image in the KITTI data set, with various degrees of occlusion and truncation. We selected 3500 images from the KITTI data set for testing, including 3000 training sets and 500 verification sets.

In this experiment, our purpose is to detect the pedestrians in the data set and compare it with the verification set, and we hope to get high accuracy in this process. We use the Git Clone instruction to clone the corresponding target detection YOLOV3 code base on Github in the first step.

Secondly, we configure the environment and choose Python 3.8 according to the compatibility of different versions of this program. Then all kinds of libraries needed for this target detection are installed one by one, such as Matplotlib library about scientific computing, Opencv library about visual recognition, etc.

In the second step, we detected the INRIA and KITTI data sets, but before this kind of step, we preprocessed the data. The so-called data preprocessing is to convert the PNG format of the tag file in the INRIA data set into the TXT format that YOLOv3 can recognize. Then we can generate the data file needed for training, and the data file can tell us the type of detection.

The third step is to complete the network model configuration file processing, represented in the CFG format. Because 80 classes are detected in the original code, but pedestrian detection is only for human detection, we change the YOLO layer's value to 1. Then we need to change the Filters in front of the YOLO layer to 18. What is else, YOLOV3 has three output layers, so all three filters should be changed to 18.

In the fourth step, we design the corresponding algorithm training process. The algorithm iterates over 100 rounds of and batch size could be set to 64. Then we adopt the multi-scale training of Multi-Scale, so we can randomly adjust the size of the input image, which can improve the robustness of the

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

program model. Finally, we cut the image to a standard rectangle, thus reducing the amount of computation of the program.

The last step is about writing the detection code, which is the verification of a result of a pedestrian detection model trained in the above steps. First of all, it is necessary to configure a network model in CFG format and then to provide a file source for detection. Because YOLOV3 has undergone five downsampling times, the value of image size needs to be set to a multiple of 32. Finally, you can start to observe the test results by configuring relevant instructions such as device, save-txt and so on.

4.2 Data processing

In the second step of the operation step, we need to preprocess the data. The so-called data preprocessing is to convert the PNG format of the tag file in the INRIA dataset into the TXT format recognized by YOLOv3. It is worth mentioning that each line represents a target in the TXT file, and each target corresponds to five values. From left to right, they are the category of detection, the proportion of the Abscissa of the centre point relative to the width of the picture, the proportion of the ordinate of the centre point relative to the height of the picture, the ratio of the width of the detection box to the width of the picture, and the ratio of the height of the detection box to the height of the picture. It is worth mentioning that these five values are all in the range of 0-1 because they are normalized. After these steps, the Data file needed for training can be generated, and the data file can tell us the type of test.

4.3 Experimental results

We detect the INRIA and KITTI data sets, respectively. The visual detection results for the INRIA data set and the KITTI data set are shown in figure 2 and figure 3, respectively. As what can be seen easily from these figures, after training, there will be a corresponding detection box for each pedestrian; this detection box probably determines the location of pedestrians, then there is a confidence value at the top of the detection box, which is a number in the range of 0-1, indicating the probability of pedestrians in the detection area. Moreover, after improving the skeleton network structure of YOLOV3, after 20 iterations of training, we input 512-512 images to compare the accuracy of the verification set. It is easy to find that the new model's accuracy has been improved to a certain extent.

First, it is shown in figure 4, which represents the changes in the values of precision, mAP@0.5, Recall, and F1 in the INRIA dataset. Among these four values, precision represents the accuracy, which indicates that the actual number of positive samples accounts for the number of positive samples that the network considers to be positive samples, so it directly reflects the accuracy of the target detection network. Secondly, Recall represents the recall rate, which indicates the proportion of the real positive samples identified by the network to the actual positive samples, which also directly reflects the accuracy of detection to some extent. Then the F1 is calculated from the values of Recall and precis, and its purpose is to locate the harmonic average of Precision and Recall. Finally, the value of mAP@0.5 can also reflect the accuracy of detection to a certain extent.

In figure 4, the values of Precision, mAP@0.5 and FI have been significantly improved, but the record value has decreased, which proves that the change of the skeleton network is practical. However, the value mAP@0.5 did not improve significantly, only increased from 0.915 to 0.921; this is because there are fewer small targets in the INRIA data set, but our change mainly in the fusion with the shallower features, which is an introduction of more low-level spatial information, it mainly improves the accuracy of small target detection, so this not reflected in the change of numerical mAP@0.5. Figure 5 shows the changes of precision, mAP@0.5, Recall and FI in the KITTI dataset. Because there are more small targets in the KITTI data set, so it better reflects the impact of changes in our YOLOv3 skeleton network on detection. Finally, the experimental results show that the four reference values of precision, mAP@0.5, Recall and F1 are all improved. The reference value mAP@0.5 is increased from 0.18 to 0.427, which shows that the accuracy of the network for small target detection is greatly improved.

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

Figure 2. The visual detection results for the INRIA data set

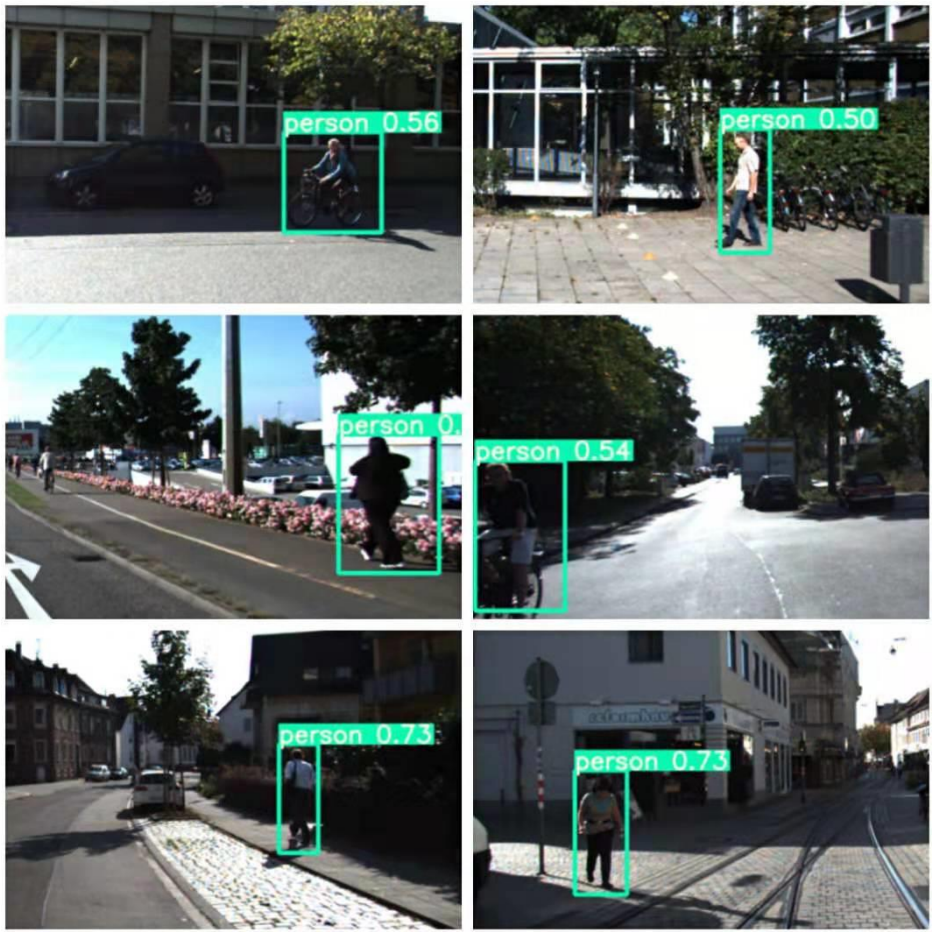

Figure 3. The visual detection results for the KITTI data set

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

Dataset Category	INRIA			
Index	Precision	Recall	mAP@0.5	F1
YOLOv3	0.834	0.891	0.915	0.862
Modified YOLOv3	0.889	0.876	0.921	0.882

Figure 4

Dataset Category	KITTI			
Index	Precision	Recall	mAP@0.5	F1
YOLOv3	0.447	0.173	0.18	0.25
Modified YOLOv3	0.616	0.425	0.427	0.503

Figure 5

5. Conclusion

This essay is based on the improvement of the YOLOv3 algorithm, an end-to-end multi-scale feature fusion target detection network is proposed for target detection in pedestrian scenes. The algorithm encodes the semantic dependence between space and channels, fuses multi-scale features, and significantly improves the accuracy of small target detection without increasing the amount of computation. In addition, we validate the algorithm on a large-scale data set covering a variety of environments and pedestrian types, and use mosaic data enhancement method to further enrich the training samples without enhancing the training time, and obtain a more accurate anchor frame through K-means clustering at the detection end, so that the location of the target area can be predicted more accurately in the subsequent regression calculation. In the experimental results, the pedestrian detection accuracy of the INRIA data set has been slightly improved. In contrast, the pedestrian detection accuracy of the KITTI data set with many small targets and a more complex environment has been significantly improved. This proves that the network enhances the expression ability of small target features, enhances the robustness of pedestrian detection in a complex environment, and dramatically improves pedestrian detection accuracy.

References

[1] Jun Hua, Zhicheng Sun, Junwei Zhao, and Tong Zhu, "Pedestrian active safety system and its impact on traffic flow," Journal of Chongqing Jiaotong University (Natural Science Edition), vol. 40, no. 03, pp. 34 -- 42.

[2] Li Ling, "Study on the Impact of Traffic Conflicts on the Traffic Safety of Pedestrians", Smart City, Vol.V.6; No.96, No. 23, pp. 133 -- 134, 2020.

[3] Yuliang Hong, Pingyi Ye, Lin Zhao, Xingya Zhang, Guanghua Zhao, "A Study on Pedestrian Operation Safety of Large Events", Transportation and Transportation, Vol. V. 33; No.48, No.S2, pp. 100 -- 104, 2020.

[4] Yan Xilei and Wang Yunxia, "Traffic Safety Problems and Improvement Suggestions of Pedestrian Crossroads in Sections", China Public Safety (Academic Edition), No. 2, 2019.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), 2005, vol. 1, pp. 886–893.

[6] P. Viola and M. Jones, "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, 2001, vol. 1, pp. I–I.

[7] P. Viola and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp. 1627–1645, 2009.

[9] A. Prioletti, A. Møgelmose, P. Grisleri, M. M. Trivedi, A. Broggi, and T. B. Moeslund, "Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 3, pp. 1346–1359, 2013.

ISSN 2616-5775 Vol. 4, Issue 5: 76-84, DOI: 10.25236/AJCIS.2021.040511

[10] K. C. Kirana, S. Wibawanto, and H. W. Herwanto, "Redundancy Reduction in Face Detection of Viola-Jones using the Hill Climbing Algorithm," in 2020 4th International Conference on Vocational Education and Training (ICOVET), 2020, pp. 139–143.
[11] J. X. Zeng and X. Chen, "Pedestrian Detection Combined with Single and Couple Pedestrian DPM Models in Traffic Scene," Acta Electronica Sinica, 2016.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.
[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
[14] R. Girshick, "Fast R-CNN," Proceedings of the IEEE International Conference on Computer Vision, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.
[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.
[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[18] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
[19] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Tech report, pp. 1–6, 2018, [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf.
[20] W. Liu et al., "SSD: Single shot multibox detector," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.
[21] J. Han, Y. Liao, J. Zhang, S. Wang, and S. Li, "Target fusion detection of LiDAR and camera based on the improved YOLO algorithm," Mathematics, vol. 6, no. 10, p. 213, 2018.
[22] P. Kuang, T. Ma, F. Li, and Z. Chen, "Real-time pedestrian detection using convolutional neural networks," International Journal of Pattern Recognition and Artificial Intelligence, vol. 32, no. 11, p. 1856014, 2018.
[23] Li Fujin, Meng Luda, "Pedestrian Detection Algorithm Based on Feature Pyramid SSD," Journal of North China University of Science and Technology (Natural Science Edition), vol.v.43; No.15, No. 01, pp. 120 -- 126, 2021
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
[26] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, "A survey of decision fusion and feature fusion strategies for pattern classification," IETE Technical review, vol. 27, no. 4, pp. 293–307, 2010.