ISSN: 2664-9640

Research on Intelligent Identification Method of Urban Management Case Based on Jetson TX2

Yongxia Yang, Xiaoyan Li, Man Wang

School of Electronics and Information Engineering, Xi'an Technological University, Xi'an 710021, China

Abstract

Smart city management is a major research hotspot in the field of artificial intelligence. Smart city management is the use of data mining, video image algorithms, scheduling algorithms and deep learning algorithms to solve urban management problems with maximum efficiency. Rely on special personnel to identify, distribute and handle cases. This makes the system inefficient, and also wastes a lot of financial and material resources. In response to this problem, this paper studies the target detection of video images of urban management cases based on the yolo algorithm, and integrates the target detection algorithm into the city management system. First, collect and label the data set, and then perform data enhancement on the existing data set, and train the processed data set in the yolo network to achieve the purpose of intelligent identification of urban management cases. The final recognition rate can reach 90%. Finally, the TX2 mobile platform is used to call the weight files obtained from the training to verify the city cases and obtain the marked pictures or videos.

Keywords

Yolov3; convolution neural network; Target detection algorithm; Jetson TX.

1. Introduction

The intelligence of city management is an important part of artificial intelligence research. It can solve city management problems in the most effective way through data extraction [1], video image algorithms, scheduling algorithms and deep learning algorithms, minimize the number of staff, and maximize Use information resources effectively [2]. In recent years, more and more cities have adopted smart city management systems to manage city operations more conveniently and quickly [3]. In addition to the well-known first-tier cities in China, most domestic second-tier cities, such as Chongqing, Qingdao, and Jinan, have become pilot cities for smart cities, and have established smart city management in certain aspects through advanced technology and innovative capabilities [4-8].

In the task of intelligence of urban management cases, there are problems such as rich and changeable urban management cases, occlusion among various cases, and complex detection environment. In daily life, each type of urban management case is different, and the manifestation of each type of urban management case will be different, and the urban management case will take a variety of attitudes[9、10]. The image background sometimes occludes and influences each other, and under the influence of external factors such as the change of light intensity in the real environment, the detection task is more difficult [11].

The use of image recognition algorithms to manage urban cases is of very important application significance. In the academic world, a wave of deep learning research has been set off, and it has become the main force in the development of artificial intelligence [12]. At the same time, the target detection algorithm is an inseparable part of the deep learning algorithm [13-15]. However, there are still some problems in the application of video image recognition in urban

ISSN: 2664-9640

management cases, which is necessary and feasible for in-depth research. Putting optimized and improved target detection algorithms and models into the city management system [16], such as intelligent recognition of urban management cases such as mobile stalls occupying roads in pictures and random parking of shared bicycles, can promote the rapid development of target detection algorithms.

Target detection integrates many advanced technologies such as image processing, model recognition, feature extraction, and deep learning, which is a challenging subject [17]. At present, many important results have been achieved in target detection, and these results have been widely used in the fields of security, industry, and vehicle driving. At the same time, it also faces many challenges, such as detection in complex and diverse background environments, strict requirements for system real-time and stability, and detection of various appearance features [18].

In China, Wang Jingyuan and others have developed urban intelligent transportation and urban computing technologies to make the city smarter and smarter through the research of different domestic urban conditions; Wen R et al[19、20]. After analyzing the current situation of Suzhou city, they proposed new The smart city construction plan under the smart power cable management mode; Xia L and others analyzed the interactive relationship between the government and the masses, as well as the technology and infrastructure that enterprises and people are more concerned about, and proposed the intelligent perception information service Smart city [21].

2. Detection algorithm based on urban management cases

2.1. Target detection algorithm based on YOLO

Yolo v3 inputs the 448×448 image directly into the neural network, runs the convolutional neural network [22], and then thresholds the model confidence card to determine the location and type of the model[23]. However, when YOLO detects small targets and targets close to each other, the detection effect is very poor[24]. Therefore, in order to cope with the problems of YOLO, the SSD algorithm framework is improved, and the RPN structure is selected. The regression mechanism of the YOLO algorithm framework has also been improved [25], that is, the category score and offset of the target frame are predicted on multiple hierarchical feature maps. Although the framework of the SSD algorithm can solve the processing of the target frame for most different levels, the detection of small targets needs to be improved. Yolov2 found a matching point between the accuracy of the test and the speed of the test [26-27], thereby maintaining a balance between the two. A joint training method for target classification and detection is proposed [28]. Through this method, more than 9000 targets in the ImageNet dataset are tested in real time. Compared with YOLOv2, YOLOv3 introduces a 53-layer residual network as a feature extractor, which improves the detection performance a lot.

Target detection based on Yolov3 algorithm is divided into two stages:

(1) Training phase. In the training phase, if the center of the object is in a range, use the object label (including x, y, w, h, and category) to mark the area. This sets the training label. That is, in the training phase, we teach the region which object in the image to predict.

(2) Testing phase. Just as you successfully learn to predict objects in this area during training, this area will do the same.

2.2. YOLO V3 network structure

The convolutional layer of YOLO V3 ranges from 0 to 74, with a total of 53 layers. The input pixels on the network are 416×416. The number of channels collected by each layer of convolution is 3.0, and the input data is processed by BN. The picture below shows the picture size set in the yolo convolutional layer:

ISSN: 2664-9640

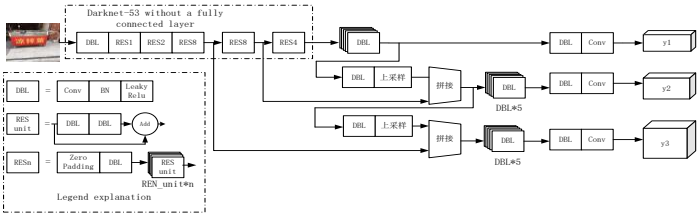

Figure 1 YOLO v3 structure

Figure 2 Enter the picture size

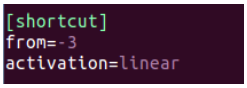

Figure 3 Shortcut

The above picture shows the shortcut content. Add the number of channels (c), the width of the picture (w), and the height of the picture (h) to the same two layers to form a layer with the same chew; from=-3: the same as the third layer from the previous add. Set the convolutional layer parameters according to your own recognition category as shown below:

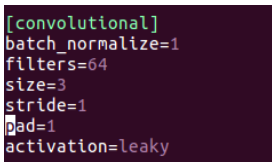

Figure 4 Convolutional layer

The content of the Yolo layer is shown in the figure below, and the parameters are explained in the table below:

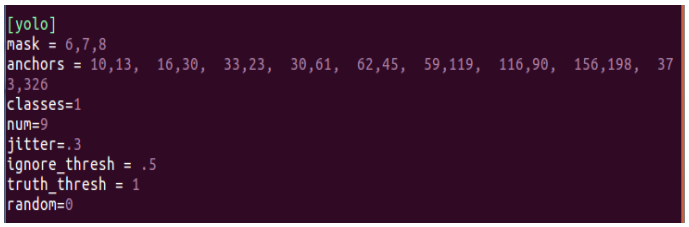

Figure 5 YOLO layer

Table 1 YOLO layer parameter meaning

parameter	Parameter meaning
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326	Pre-selection box, the value calculated by the sample

ISSN: 2664-9640

mask = 6,7,8	Which preselection box currently belongs to
classes=1	The number of types of objects that the network needs to recognize
num=9	The number of preselected boxes, that is, the total number of anchors
jitter=.3	Increase noise by dithering to suppress overfitting

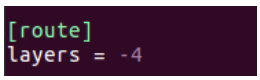

Figure 6 Circuit layer

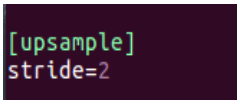

Figure 7 Up-sampling layer

The above two pictures are the circuit layer and the up-sampling layer in the yolo algorithm.

3. Experiment content

According to the needs of the task, this experiment is divided into seven steps:

(1) Data set collection: Search and download the required data set in the browser.

(2) Data set mark: Select the data set and mark the target location.

(3) Data enhancement: used to improve the generalization ability of the training set and reduce overfitting.

(4) VOC data set production: the data set is produced in an appropriate format, and the python script is run to generate training and verification sets, which is convenient for training the data set.

(5) Training data set: to extract image characteristics and get the average loss.

(6) Test data set: test training results.

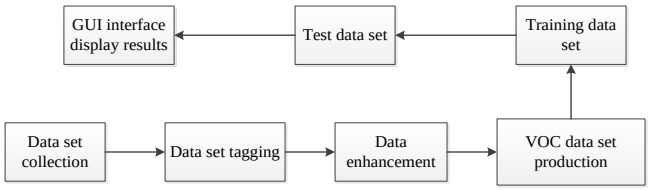

Figure 8 Experimental steps of target detection

3.1. Data set collection

Python 3.5 replaces the system default Python 2.7, install the software Anacanda (python3.5 corresponds to the version Anadanca4.2.0). The following interface appears after the installation is successful:

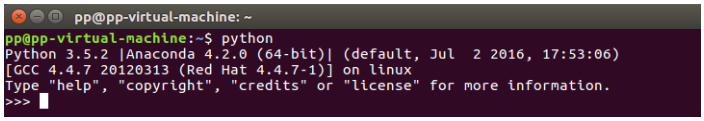

Figure 9 Installation Anadanca

Search the mobile booth pictures in the Firefox browser under the Ubuntu system, and then sort the photos. In order to enhance the generalization ability of the network model and prevent the trained network from over-fitting, the image samples are used as training data before input into the network by random Rotating and translating the image and changing the saturation,

ISSN: 2664-9640

exposure and hue of the image will not only get more samples, but also enhance the network model's ability to judge new data samples. The collected data sets are as follows:

Figure 10 Data set collection

3.2. Data set tagging

Download the Labeling software, use the terminal to compile and install the software. After the software is installed successfully, use the python labelimg.py command in the Labeling-master directory to open the labeling software as follows:

Mark all the data sets collected and mark a few if there are several targets to ensure that each target can be successfully marked. The marked pictures and files are as follows:

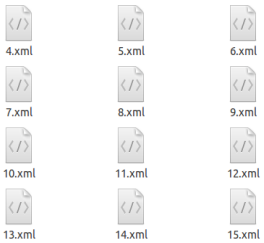

Figure 11 The generated .xml file

3.3. Data enhancement

On the basis of existing data sets, perform data augmentation. For image data, there are many data augmentation methods that can be done. The usual methods are: translation, rotation, scaling, cropping, and shearing, horizontal/vertical flip. The picture after flipping, rotating and cutting is as follows:

Figure 12 Flip horizontally

ISSN: 2664-9640

Figure 13 Rotate Left and Cut Right

After data enhancement, although the accuracy rate on the training data set has decreased, the accuracy rate on the verification data set has been significantly improved, indicating that the generalization ability of the model has been enhanced.

3.4. VOC data set production

Download Darknet from the browser. Create a VOC2020 folder in the Darknet folder and store the original pictures in the JPEGIANES folder. Place the XML file generated by the markup in the Annotations folder and create a new folder named Main. Use python script to generate four files train.txt, val.txt, test.txt and trainval.txt under this folder:

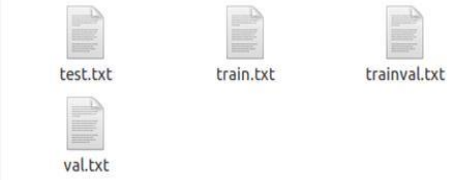

Figure 14 Four generated txt files

Then create a labes.py file in the Darknet folder, and generate a labels folder in the Voc2020 directory after running, in which the image txt file generated from the original image stored in the Voc2020 folder is stored, which stores all the training sets generated Absolute path. At this point, the VOC data set has been produced. The contents of the generated VOC2020_train.txt file and labels folder are as follows:

```
/home/pp/darknet/voc2020/JPEGImages/16.jpg
/home/pp/darknet/voc2020/JPEGImages/221.jpg
/home/pp/darknet/voc2020/JPEGImages/22.jpg
/home/pp/darknet/voc2020/JPEGImages/120.jpg
/home/pp/darknet/voc2020/JPEGImages/238.jpg
/home/pp/darknet/voc2020/JPEGImages/178.jpg
/home/pp/darknet/voc2020/JPEGImages/65.jpg
/home/pp/darknet/voc2020/JPEGImages/92.jpg
/home/pp/darknet/voc2020/JPEGImages/214.jpg
/home/pp/darknet/voc2020/JPEGImages/276.jpg
/home/pp/darknet/voc2020/JPEGImages/269.jpg
/home/pp/darknet/voc2020/JPEGImages/239.jpg
```

Figure 15 The generated VOC2020_train.txt file

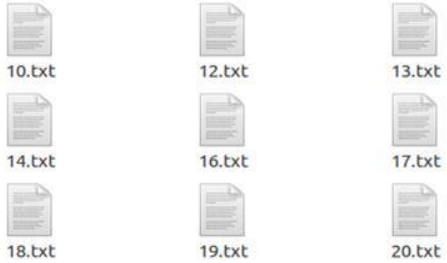

Figure 16 The generated labels folder

The following figure shows the content of the generated VOC2020_train.txt file:

ISSN: 2664-9640

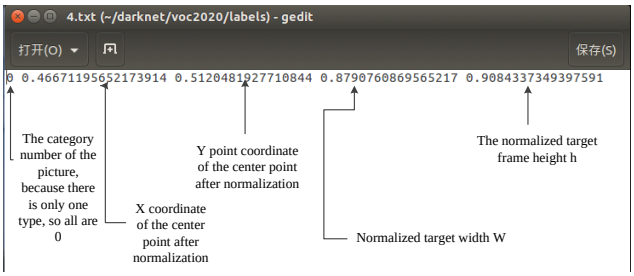

Figure 17 Parameters in the VOC2020_train.txt file

3.5. Yolo-v3 training data set

Modify the configuration file:

```
classes= 1
train  = /home/pp/darknet/voc2020/voc2020_train.txt
names = /home/pp/darknet/voc2020/voc2020.names
backup = /home/pp/darknet/voc2020/weights
```

Figure 18 My_data.data file

The following figure shows the my_yolov3.cfg file and VOC2020.names file that need to be modified:

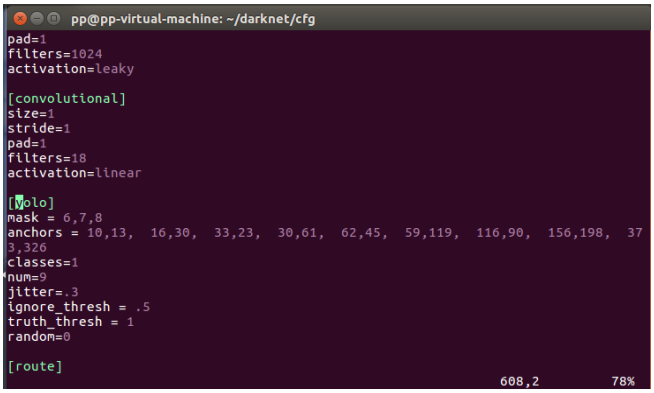

Figure 19 My_yolov3.cfg file

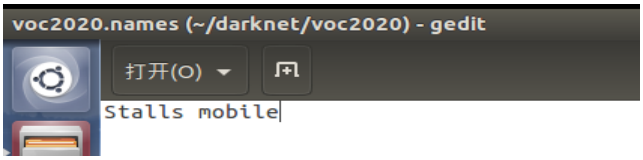

Figure 20 VOC2020.names file

Training data set: Download pre-training weights, and start training after downloading. The training data set commands are as follows:

```
pp@pp-virtual-machine:~/darknet$ ./darknet detector train cfg/my_data.data cfg/my_yolov3.cfg darknet53.conv.74
```

Figure 21 Training data set instructions

Save the training log. After the training is over, a weight file will be created. As shown below:

ISSN: 2664-9640

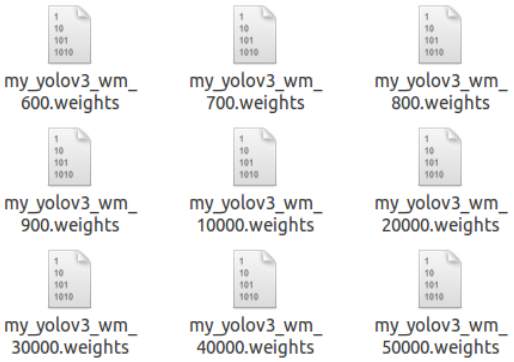

Figure 22 Weight file

3.6. Test data set

Change the python program in the yolov3.cfg file to test mode. Open the Ubuntu terminal and enter the test data set command, where the file name uses the path you set. The test data set commands are as follows:

```
pp@pp-virtual-machine:~$ ./darknet detector test cfg/my_data.data cfg/my_yolov3.
cfg voc2020/weights/yolov3-voc_20000.weights voc2020/JPEGImages/29.jpg
```

Figure 23 Test instruction

Figure 24 Test results

4. Experimental results and analysis

4.1. Training log and its analysis

The training log should be saved during the training data set. The training command to save the training log is as follows:

```
pp@pp-virtual-machine:~$ cd darknet
pp@pp-virtual-machine:~/darknet$ ./darknet detector train cfg/my_data.data cfg/m
y_yolov3.cfg darknet53.conv.74 | tee train_yolov3-voc.log
```

Figure 25 Generate training log

It can be seen from the training log that the number of occurrences of nan is about 30%, which is a normal value, and neither the total loss nor the average loss shows nan, so it is inferred that there is no error in the training. The saved training log is as follows:

ISSN: 2664-9640

```
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000007, .5R: -
nan, .75R: -nan,  count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000002, .5R: -
nan, .75R: -nan,  count: 0
50197: 0.160969, 0.119643 avg, 0.000000 rate, 0.045360 seconds, 50197 images
Loaded: 0.000030 seconds
Region 82 Avg IOU: 0.638947, Class: 0.999179, Obj: 0.991913, No Obj:
0.007262, .5R: 1.000000, .75R: 0.000000,  count: 1
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000007, .5R: -
nan, .75R: -nan,  count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000003, .5R: -
nan, .75R: -nan,  count: 0
50198: 0.094352, 0.117114 avg, 0.000000 rate, 0.045080 seconds, 50198 images
Loaded: 0.000032 seconds
Region 82 Avg IOU: 0.727500, Class: 0.999343, Obj: 0.965018, No Obj:
0.002828, .5R: 1.000000, .75R: 0.000000,  count: 1
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000022, .5R: -
nan, .75R: -nan,  count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000002, .5R: -
nan, .75R: -nan,  count: 0
50199: 0.079194, 0.113322 avg, 0.000000 rate, 0.045152 seconds, 50199 images
Loaded: 0.000031 seconds
Region 82 Avg IOU: 0.893102, Class: 0.999834, Obj: 0.999469, No Obj:
0.002034, .5R: 1.000000, .75R: 1.000000,  count: 1
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000008, .5R: -
nan, .75R: -nan,  count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.000002, .5R: -
nan, .75R: -nan,  count: 0
50200: 0.021589, 0.104149 avg, 0.000000 rate, 0.045104 seconds, 50200 images
```

Figure 26 Training log

4.2. Recognition results and their comparison

In this paper, the target detection algorithm based on yolo v3 is used for target recognition. The number of targets in the acquired data set is different, the position and size are different, and the accuracy of recognition is also different.

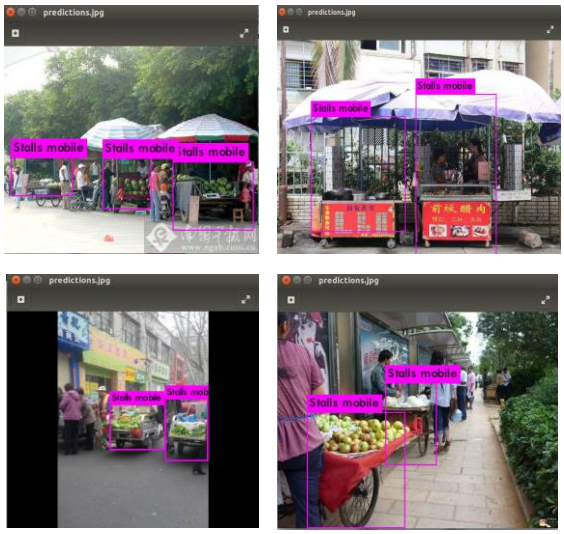

Figure 27 Multiple recognition targets

4.3. Jetson TX2 porting results

Jetson is a low-consumption embedded development platform, whose modules can be used for large-scale applications with different performance and price levels. It is equipped with Linux16.04 operating system, which makes it very convenient to use.

Create a new folder, go to the official website to download JetPack, configure the relevant environment for it, update the relevant files and save them under the created directory. Select the environment under Target-Jetson TX2/TX2i to configure TX2. Create user name and password, wait for configuration After basic configuration of TX2, Linux16.04 interface will appear. TX2 is shown in the figure below:

ISSN: 2664-9640

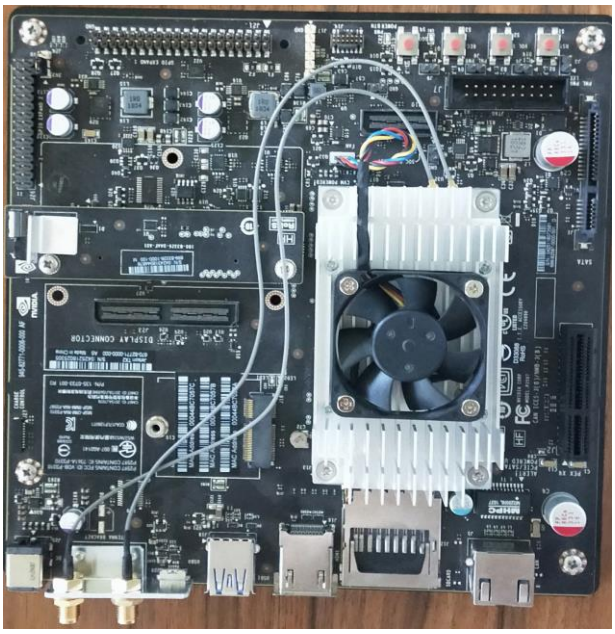

Figure 28 Jetson TX2

Use USB to copy the weight file, related configuration file and original image from the computer, correct it to test mode on TX2, install the Canda and Opencv software required by GPU. Modify the Makefile under the darknet folder to GPU and then test.

5. Conclusion

The YOLO algorithm has good real-time performance and high recognition accuracy, and is widely used in target detection based on deep learning. But it is undeniable that the YOLO algorithm has certain limitations. Since each cell can only predict 2 boxes, the IOU height of gt_box is taken as the final detection frame for each cell, which means that each cell can only predict at most one target. If a single cell contains multiple targets, the algorithm can only detect one of them, which will result in missed detection of small targets. Therefore, YOLO has low detection accuracy for small target objects and images with denser targets. And because the output layer is a fully connected layer, during detection, the YOLO training model only supports input images with the same resolution as the training image; although YOLO can reduce the probability of detecting the background as an object, it will also lead to a lower recall rate. After updating the version to YOLO V3, although the recognition effect of small targets is enhanced, there are still some shortcomings.

In using the YOLO algorithm to detect the model, it is necessary to carefully prepare the data set to be used. The quality of the data set will directly affect the later training results. Therefore, the following points should be considered as much as possible during the data set production process:

1) The number of pictures in the data set should be as large as possible. When training with more data sets, different characteristics will be obtained, making the results more accurate.

2) The pictures in the data set should be collected under different lighting conditions and different weather conditions to improve the ability to distinguish backgrounds.

3) Use data enhancement technology to improve the generalization ability of the model and reduce the phenomenon of over-fitting.

4) The data set save path should be clear enough to prevent the corresponding path from being found later.

5) Make a standardized data set to prevent it from being unopened or unloaded during training.

ISSN: 2664-9640

Acknowledgements

This work was supported by the Scientific Research Program Funded by Xi'an City Weiyang District Science and Technology Department of China (201923) and Science and Technology Program of Xi'an Science and Technology Department (No. 2020KJRC0037).

References

[1] BYOUNG C K,KWANG H. Fire detector based on vision sensor and support vector Machines [J] Fire Safety Journal,2009,44;322-329.
[2] Redmon J,Divvala S,Girshick R,et al.You Only Look Once: Unified, Real-Time Object Detection[C]. Las Vegas: IEEE International Conference on Computer Vision, 2016.
[3] Editorial Department of "Chinese Administration". Continuously create new urban development bureaus
Surface [J]. Chinese Administration, 2016(2).
[4] Lecun Y,Bottou L,Bengio Y,et al.Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998,86(11): 2278-2324.
[5] YAVUZ A, KAGAN T B. Deteting interferences with ios applcations to me a sure speed ofsound [J]. Physics Education, 2016, 51(1)：015009.
[6] Zhang H , Du Y , Ning S , et al. Pedestrian Detection Method Based on Faster R-CNN[C]. International Conference on Computational Intelligence & Security. IEEE ComputerSociety, 2017.
[7] Huang Kaiqi, Chen Xiaotang, Kang Yunfeng. Summar of Intelligent Video Surveillance Technology[J]. Chinese Journal of Computers, 2015, 20(6)1093-1118.
[8] Wang Yulong. Research on GPU-based video big data processing method [D]. Dalian: Dalian University of Technology, 2017.
[9] Girshick R.Fast R-CNN[C].Santiago:IEEE International Conference on Computer Vision.2015.
[10] Wu Xiyu. Deepen the reform of urban management and build a livable Nanning[N]. Guangxi Daily, 2015-12-30. (007).
[11] Du Lijuan, Lu Xiaoya. Intelligent positioning and tracking of multi-view targets in video surveillance Trace method research [J]. Science Technology and Engineering, 2017, 17(16): 270-274.
[12] BAKICI T,ALMIRALL E, WAREHAAM J. A Smart City Initiative: the Case of Barcelonal[J]. Journal of the Knowledge Economy,2013,4(2):135-148.
[13] Dan Xu, Wanli Ouyang, Elisa Ricci, et al.Learning Cross-Modal Deep Representations for Robust Pedestrian Detection[C].Honolulu,Hawaii: The IEEE Conference on Computer Vision and Pattern Recognition,2017.
[14] Fu Wei, Wang Jinqiao, Teng Kezhen. Surveillance video project based on deep learning Index search [J]. Radio Engineering, 2015, 45 (12): 16-20.
[15] Zhang Wanpeng. Research on the people flow statistics system of dense crowds [D]. Mountain East: Shandong Normal University, 2018.
[16] Bao Ling. Research and system of intelligent people counting technology based on deep learning System design [D]. Sichuan: University of Electronic Science and Technology of China, 2018.
[17] Shanshan Zhang,Jian Yang,Bernt Schiele.Occluded Pedestrian Detection Through Guided Attention in CNNs[C].Salt Lake City,Utah:The IEEE Conference on Computer Vision and Pattern Recognition,2018.
[18] Du Lijuan, Lu Xiaoya. Intelligent positioning and tracking of multi-view targets in video surveillance Trace method research [J]. Science Technology and Engineering 2017, 17 (16): 270-274.
[19] Meng Fanyang. Several Key Technologies in Multi-camera Panoramic Video SurveillanceResearch [D]. Shenzhen: Shenzhen University, 2016.
[20] Girshick R,Donahue J,Darrell T,et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C].Columbus,OH:IEEE Conference on Computer Vision and Pattern Recognition,2014.
[21] Zhifeng Zhao,Shaoren Zheng.Research on Ad-hoc Network Architecture[J].Telecom Section Science, 2001, 17 (1): 14-17.

ISSN: 2664-9640

[22] Long Yu. Research on moving target detection and target tracking under occlusion conditions [D]. Wuhan: Wuhan University of Science and Technology, 2018.52.
[23] LOWE D G. Distinctive Image Feature From Scale-invariant Keypoints [J].International Journal of Computer Vision, 2004,60(02):91-110.
[24] GIRSHICK R. Fast R-CNN[C]. Proceedings of IEEE International Conference on Computer Vision.Piscataway, NJ: IEEE, 2015:1440-1448.
[25] Hosang J, Omran M, Benenson R et al. Taking a deeper look at pedestrians[C].Boston,MA: IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[26] Ding Wenxiang, Zhu Kongfan. Real-time video download based on convolutional neural network Human detection[J]. Information Technology, 2016,12:44-47.
[27] Ren S,He K,Girshick R,et al.Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.
[28]K.he,X.zhang,S.Ren,and J.Sun.Spatial pyramid pooling.indeep convolutional networks for visua lrecognition[C].European.Conf.Comput.Vision,2014:346-361.