Design of People Flow Monitoring System in Public Place based on MD-MCNN

Xiong Tianmin1, Lei Xiaochun※1,2, Zheng Xingwen1, Chen Junyan1,3, Feng Yizhou1

1School of computer and information security, Guilin University of Electronic Technology, Guilin 541004, China;
2Guangxi Colleges and Universities Key Laboratory of Intelligent Processing of Computer Images and Graphics, Guilin 541004, China;
3Guangxi Cooperative Innovation Center of cloud computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China)
※Corresponding author’s e-mail: glleixiaochun@qq.com

Abstract. Due to the limitation of hardware resources, the traditional people flow monitoring system based on computer vision in public places can’t meet different crowd-scale scenarios. Therefore, a people flow monitoring system based on MD-MCNN algorithm is designed, which is an application system combining the improved SSD object detection algorithm MNSSD and MCNN density map regression algorithm. In the initial stage, the system uses MNSSD for accurate detection and counting. If the people flow gradually reaches a certain threshold, the system automatically uses MCNN to estimate people flow until the people flow falls below the threshold. Through the experimental verification, the system can realize the people flow statistics of low-density and high-density people in different scenarios, and can be applied on the existing embedded platform. This scheme can be extended to smart cities, smart scenic spots, smart transportation and other fields.

1. Introduction

With the improvement of economic level, the acceleration of urbanization process, the increase of large-scale events, holiday scenic spots, public gatherings and other explosive crowd gathering activities, the demand for early warning of people flow is also increasing. The stampede on the Bund, illegal parade, crowds during the outbreak of COVID-19, etc., highlight the prevention of and sounds an alarm to public safety and human wealth under extreme conditions.

The current commonly used pedestrian detection and pedestrian counting systems are mostly based on a single object detection or regression estimation method, only for small or large numbers of people, and cannot cover complex crowd-scale scenarios. For instance, the pedestrian detection algorithm is still used when people flow is large, but the existing edge-end platforms have insufficient computing power. In addition, in most scenarios where people flow is not large, the MCNN algorithm is prone to problems of insufficient accuracy, resulting in false alarms[1]. In view of the above problems, a set of public location monitoring system is designed that can be applied to a variety of crowd-scale scenarios. It also proposes to use the improved Single Shot MultiBox Detector[2] (SSD) algorithm and Multi-column Convolutional Neural Network[3] (MCNN) algorithm to apply to the low computing power embedded platform for the detection of people flow MobileNet-SSD&MCNN (MD-MCNN) algorithm.
2. The implementation of the system

2.1. The System's overall framework
The system consists of four parts: camera terminal, MNSSD processing branch, MCNN processing branch, and visual monitoring terminal. The design frame diagram of the system is shown in Figure 1.

![Design frame diagram of the system](image)

Figure1. The design frame diagram of the system

2.2. The System’s application process
(1) The camera terminal acquires video streams, and the MNSSD processing branch is set by default.
(2) The MNSSD processing branch performs head detection to accurately statistics people flow. If people flow is greater than the threshold, the system switches to the MCNN processing branch.
(3) MCNN processing branch performs density map regression to estimate people flow. When people flow is less than the threshold, the system switches back to the MNSSD processing branch.
(4) The detection maps, density maps and number of people information obtained from the MNSSD and MCNN processing branches present visual charts such as line charts, histograms, and roulette charts through the visual monitoring layer.

3. The implementation of MD-MCNN algorithm
The traditional people flow statistical scheme mainly uses a single object detection algorithm or a density map regression algorithm.

In view of the complementary advantages and disadvantages of the crowd statistics algorithm for object detection and density map regression described in the introduction, this paper has designed a crowd statistics algorithm suitable for a variety of crowd density scenarios. By setting the threshold, the algorithm selects the improved SSD object detection algorithm MNSSD (MobileNet-SSD) in order to obtain a more accurate number of people when people are not large. It can accurately locate the pedestrian position and can be used for detection other crowd behaviors such as running, gathering, fighting. In scenarios where there are a large people flow and crowds gather, the algorithm switches to the crowd number estimation algorithm MCNN based on density map regression to obtain the crowd density map and estimated people flow. At the same time, it can also understand the distribution of the crowd according to the crowd density and warn of abnormal aggregation in advance event. In this solution, the algorithm switching part can be set by a threshold or adaptively according to the computing power at the edge.

3.1. The implementation of the object detection algorithm using MNSSD
The classic SSD network is an object detection network based on the Visual Geometry Group (VGG) feature extraction network\(^4\). According to the actual application scenario, this paper uses MobileNet\(^5\)network to perform feature extraction to improve the performance of traditional VGG-SSD, and improve it into a MNSSD algorithm that is more suitable for deployment to lightweight embedded devices.

3.1.1 The model structure of MNSSD
The overall framework can be interpreted as follows: after a basic feature extraction by the core
network, a multi-dimension prediction will be made based on iterating layers of the feature extraction network. The SSD feature maps of different layers are independent of each other. Conv5 and deeper layers are fed into the detection layer. The shallower layers contain mainly information for small object detection, while the deeper layers contain global information for larger object detection.

The SSD Multi-dimension feature algorithm formula is as follows[6]:

\[ T_n = S_n(T_{n-1}) = S_n \left( S_{n-1}(\ldots S_1(I)) \right) \]

(1)

\[ R = D(d_n(T_n), \ldots d_{n-k}(T_{n-k})), n > k > 0 \]

(2)

Where \((T_n)\) is the feature map of the n-th layer. \((S_n)\) is the n-th layer feature map created from the \((n-1)\)-th layer map. \((S_1(I))\) is the first layer feature map created from the image input \((I)\). \((d_n(.)\)) is the detection result on the n-th layer. \((D(.))\) is the aggregation of all intermediate values[7].

This system is built to be deployed to a lightweight embedded platform for real-time human flow detection. The lightness of the model and the impact of running speed are very important. If VGG and other models are used for training, the obtained model is very large[8]. This paper uses MobileNet network for feature extraction can improve the performance of VGG-SSD. In addition, the size of the detected object (head) is small, and the deeper features are not important, so this algorithm deletes the deep 1x1 feature map of VGG-SSD to reduce the amount of calculation. The network structure of MNSSD in Figure 2.

3.1.2 Loss function
The loss function used in this model has two parts: the loss of classification \((L_{\text{conf}})\), and the loss of position \((L_{\text{loc}})\). The total loss function is

\[ L(x, c, l, g) = \frac{1}{N} \left( L_{\text{conf}}(x, c) + \beta L_{\text{loc}}(x, l, g) \right) \]

(3)

Where \((L)\) is the total loss, \((N)\) is the number of matching default boxes. \((\beta)\) is a scalar for weighting between two-loss functions with a default value of 1.

3.2. The implementation of the density map regression algorithm using MCNN
MCNN is based on CNN (convolutional neural network), combined with multi-level receptive field and self-organized multi-layer neural network model.

3.2.1 The model structure of MCNN
The MCNN uses a 1x1 convolutional layer instead of a fully connected layer to avoid distortion. This is the advantage of using a FCN[9](Fully convolutional network). After inputting any size image, the FCN uses a deconvolution layer to up sample the last convolutional layer function map to restore the
function map to the same size as the input image, without changing the image space information. In order to achieve the purpose of predicting each pixel, the up-sampled feature map is classified pixel by pixel. In MCNN, density mapping uses filters of different sizes, and the parameters used by the convolution function are 9*9, 7*7, and 5*5, corresponding to different filter sizes.

The network structure of MCNN in Figure 3.

The key point of MCNN is the density mapping based on geometric adaptive kernel that convert the head label of the training data into a map with spatial density information of the crowd. The density is related to the people flow in a single image. The way to obtain the density map mapping is as follows:

\[ F(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_\sigma(x) \]  

In the labeled position of the head of 1 to N in a picture, the delta function of \( x \) and \( x_i \) is expressed as \( \delta(x - x_i) \), The density \( F \) corresponding to the pixel \( x \) is proportional to the convolution \( \delta(x - x_i) \), and is also positively related to the gaussian kernel \( G_\sigma(x) \), where \( \sigma \) is the width parameter.

The relationship between the density map and the pixel is expressed as (5), \( \Delta(x - x_i) \) means that there is a label at the pixel \( x_i \), \( H(x) \) means the image marked with N people. Convert (5) into a Gaussian kernel function. The Gaussian kernel is represented by the expansion parameter \( \sigma \) in (4), so the density representation is shown in (6).

\[ H(x) = \Delta(x - x_i) \]  
\[ f(x) = h(x) * \sigma(x) \]

3.2.2 Loss function

In the training phase, MCNN uses square loss to measure the distance between the estimated density map and the true density map. The square loss function is defined as follows:

\[ L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} ||F(X_i; \theta) - \bar{F}_i||_2^2 \]

Where \( \theta \) represents the parameter set to be learned by MCNN, \( N \) is the number of training images, \( F(X_i; \theta) \) represents the estimated real density map of image \( X_i \), \( \bar{F}_i \) represents the true density map of image \( X_i \), \( L(\theta) \) represents the loss between the estimated density map and the true density map. Both \( L(\theta) \) and \( F(X_i; \theta) \) are functions related to the \( \theta \) parameter set.

4 System test and results

4.1 Experimental environment

| Table 1. Model training environment | Table 2. Test environment |
|------------------------------------|---------------------------|
| Component  | Configuration          | Component  | Configuration          |
| CPU       | i7-7700HQ@2.80GHz       | CPU        | ARM® Cortex®-A57       |
4.2. Data processing

4.2.1 SSD’s data acquisition and enhancement

Because there are not many data sets for head detection on the Internet at present, we have collected thousands of pictures of different situations and numbers of people, and carried out data enhancement methods such as horizontal flipping, random cropping, and random disturbance of brightness. Data labeling is carried out through the open source software labelIMG and saved in the PASCALVOC [10] format. The Figure 4 is part of the dataset of MNSSD.

![Figure 4. The dataset of MNSSD](image)

Table 3. The structure of ShanghaiTech

| Dataset  | Count | Min  | Max   | Ave  |
|----------|-------|------|-------|------|
| ShanghaiTech Part_A | 482   | 33   | 3139  | 501  |
| Part_B   | 716   | 9    | 578   | 123  |

4.2.2 The dataset of MCNN

The MCNN training data set uses ShanghaiTech[3], which contains 1198 images and 330,000 accurately labeled head image markers. The image perspectives are different. We mainly use the picture counting of large-scale crowds to deal with possible public events. The big problem. The data set structure is shown in Table 3.

During model training, Part_A is divided into 300 training sets and 182 test sets; Part_B is divided into 400 training sets and 316 test sets.

4.3. Model training

After the data was marked, it was randomly scrambled and distributed to the training set, verification set and test set in a ratio of 8:1:1.

The prediction frame classification loss of model training and the loss of prediction frame position regression are shown in Figure 5. In the 40,000 iterations, the overall trend was declining, showing that the overall network is effective.

![Figure 5. The loss function of MNSSD](image)

![Figure 6. The loss function of MCNN](image)
MCNN model training uses weakly supervised learning\cite{11}. According to the cross-validation method, 10% of the training data is randomly selected to determine model parameters and the optimal model is selected. Every 2 iterations, H5PY is used to store data similar to arrays and directory containers. It can be seen from Figure 6 that as the total number of training sessions increases, the overall trend of the loss curve is declining, and eventually it can converge.

4.4. Experiment simulation
Connect the trained model to the system and deploy it to the terminal recognition test environment. Test the statistical effect of the system on the sparse crowd image and dense crowd image many times.

| Actual number | MNSSD detection accuracy | MCNN estimated accuracy |
|---------------|--------------------------|-------------------------|
| 0~50          | 95.875%                  | 87.233%                 |
| 50~100        | 93.463%                  | 89.327%                 |
| 100~150       | 86.354%                  | 94.345%                 |
| 150~200       | 70.782%                  | 94.424%                 |

4.5. Conclusion
According to Table 4, in the high-density crowd scenario (100~200), the pedestrian head and shoulder features are very small, and the features are almost undetectable using the MNSSD algorithm, the accuracy is extremely low, and the MCNN density map is used for regression estimation. The average accuracy rate of the number of people is 94.385%. In scenarios with sparse crowds (0~100), the average accuracy of MCNN is only 87.28%. The average accuracy of switching to MNSSD object detection algorithm is 94.669%. Therefore, we choose 100 as the threshold. When people flow estimated by the MCNN processing branch is above 100, the system uses the MCNN processing branch to estimate people flow. On the contrary, the system uses MNSSD processing branch to select pedestrian feature to obtain the accurate people flow.

5. Summary and prospects
Based on the switching between MNSSD and MCNN algorithms, an MD-MCNN algorithm is
designed and uses it to implement a set of people flow monitoring system. Experiments show that compared with other schemes, the system's people flow statistics scheme can guarantee sufficient accuracy and speed on the edge of lower computing power. And can detect and count from zero to thousands of people, covering different crowd density scenarios.

Although this system has completed the main functions, its MD-MCNN algorithm still needs to be optimized. For extreme environments such as low illumination, strong light, or fog, it is necessary to further increase the data set, test and adjust the algorithm.

Acknowledgement
This work is supported by Project to Improve the scientific research basic ability of Middle-aged and Young Teachers (No.2019KY0222,2020KY05033), Student's Platform for Innovation and Entrepreneurship Training Program under Grant (No.201910595039)

References
[1] Lin, Z.R. (2018) Embedded Station Passenger Counting System Based on Convolutional Neural Network. Xiamen University.
[2] Liu, W., Anguelov, D., Erhan, D., et al. (2016) SSD: Single shot MultiBox detector. In: European Conference on Computer Vision. Amsterdam. 21-37.
[3] Zhang, Y., Zhou, D., Chen, S., et al. (2016) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas. 589—597.
[4] Simonyan, K., Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556
[5] Howard, A.G., Zhu, M., Chen, B., et al. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861
[6] Ren, J., Chen, X.H., Liu, J.B., et al. (2017) Accurate Single Stage Detector Using Recurrent Rolling Convolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Hawaii. 752-760.
[7] Ren, Y.J, Yang, J, Liu, F.T. (2019) Research on Object Detection Method Based on SSD and MobileNet Network. Journal of Frontiers of Computer Science & Technology, 13(11):1881-1893.
[8] He, C.T, Zhu, M. (2019) No-reference Face Image Quality Assessment Based on Deep Learning for Mobile Devices. J Chin Mini-Micro Comput Syst,40(02):407-412.
[9] Long, J., Shelhamer, E., Darrell, T. (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 3431-3440.
[10] Everingham, M., Van Gool, L., WilliamsC, K.I., et al. (2013) The PASCAL Visual object classes challenge 2011 (VOC 2011) Results. http://www.pascalnetwork.org/challenges/VOC/voc2011/. 2011.
[11] Wang, H.X., Liu, P.Z., Luo Y, M. (2018) Convolutional neural network tracking algorithm accelerated by Gaussian kernel function. CAAI T Intell Syst,13(03):388-394.
[12] Biswas, D., Su, H.B., Wang, C.Y., et al. (2019) An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD. Physics and Chemistry of the Earth,110.