The 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)
May 18-20, 2018, Nanjing, China

People Counting and Pedestrian Flow Statistics Based on Convolutional Neural Network and Recurrent Neural Network

Jie Zhu
School of Information Science and Technology, Donghua University
Shanghai, China
zhujie245@foxmail.com

Fan Feng
School of Information Science and Technology, Donghua University
Shanghai, China
fengfan@mail.dhu.edu.cn

Bo Shen
School of Information Science and Technology, Donghua University
Shanghai, China
bo.shen@dhu.edu.cn

***Abstract*—People counting and pedestrian flow statistics are challenging tasks because of the perspective distortions, appearance changes and occlusion. In this paper, we address the two tasks: people counting in images of highly dense crowds and pedestrian flow statistics in a place over a period of time. Our first contribution is to propose a new convolution neural network (CNN) model which is composed of a deep and shallow fully convolution network to fulfill the task of people counting. We extract different layer features from the deep fully convolution network and the last layer features from the shallow fully convolution network, and concatenate them together. After that we add two deconvolution layers to make the output image have the same resolution with the input image. Our second contribution is to combine pedestrian flow statistics task with people counting task. According to the density maps that CNN model generates, we can calculate the number of people crossing a place based on the recurrent neural network (RNN). Besides, we also have collected two datasets and labelled them. Extensive experiments have been implemented, our people counting method outperforms other existing methods, and our pedestrian flow statistics method combined with CNN model also outperforms the model which only uses long-short term memory (LSTM).**

***Index Terms*—people counting, convolution neural network, pedestrian flow statistics, recurrent neural network**

I. Introduction

Nowadays, people counting and pedestrian flow statistics are one of the hottest research areas in computer vision. With the growth of population, the occurrence frequency of the terrorist incidents or stampedes increase. Therefore, it is crucial to obtain the number of people and analyze the pedestrian flow through video surveillance and thus recognize the abnormal behaviors in crowd. The people counting task can be achieved by counting the number of people in each frame, meanwhile, the pedestrian flow statistics can be analyzed by calculating the number of people in escalator every few seconds. The combination of these two tasks is helpful for quantifying the seriousness of accident and hence managing the flow of crowd better.

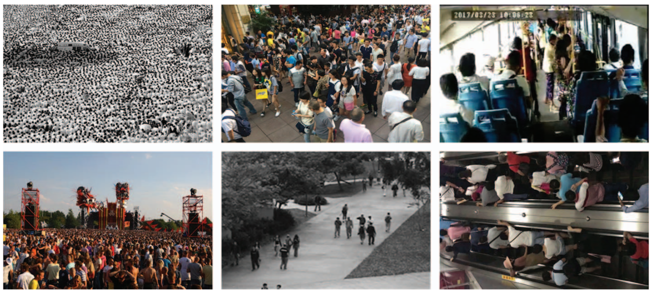
Fig. 1. Examples of density images with the drawbacks of occlusion, target deformation and illumination.

The density image plays an important role in people counting and pedestrian flow statistics. However, the density image itself has many drawbacks such as occlusion,target deformation and illumination change,see the typical examples shown in Fig. 1. As such, it is often the case that the features of the density image is not easy to extract, which may significantly affect the achievement of the people counting and the pedestrian flow statistics tasks.

In fact, for the feature extraction, there have been a number of advanced approaches available in the existing literature. Recently, some feature extraction approaches have been applied in the people counting problem and the accuracy of people counting has been improved greatly. For example, the convolution neural network (CNN), as an effective tool for feature extraction, has been successfully applied in people counting, see e.g. [1]–[3].

As for pedestrian flow statistics, it counts the number of people passing through a place over a period of time by extracting the features of successive images. A number of methods which is used to better extract the features of successive images have been proposed. Among them, RNN is the best tool for successive images features extraction.

In this paper, a new CNN model is proposed to fulfill the task of people counting in each image, then we use the output density maps that the CNN model generates to accomplish

This work was supported in part by the National Natural Science Foundation of China under Grant 61473076 and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning of China.

978-1-5386-7255-6/18/$31.00 ©2018 IEEE

the task of pedestrian flow statistics in successive images. On the one hand, in the task of people counting, we adopt a combination of deep and shallow fully convolution networks to predict the number of people. The deep fully convolution network is based on VGG-16 [4], whose fully connected layers are removed, and we extract different layer features of the deep fully convolution network. Differently, the shallow fully convolution network is a CNN of only three layers and we extract the last layer features of the shallow fully convolution network. After that, we concatenate the features which are extracted in deep and shallow fully convolution network respectively. Then we add two deconvolution layers after the concatenating features, which makes the output density map and the input image have the same resolution. On the other hand, in the task of pedestrian flow statistics, we can acquire a series of density maps according to the CNN model discussed above, then the density maps are put into the RNN to count the number of people passing through a place over a period of time.

In the experimental process, we have collected the DH302img and DH302vid datasets in Shanghai Metro during rush hours. We use DH302img and UCF_CC_50 [5] to evaluate the proposed CNN model, and we use DH302vid to evaluate the method of pedestrian flow statistics. This paper is organized as follows. In Section 1, we discuss the introduction. In Section 2, we describe related works. The details of the people counting and pedestrian flow statistics are described in Section 3. In Section 4, we evaluate our methods in DH302img, UCF_CC_50 and DH302vid. In the end, we describe our conclusion in Section 5.

II. Related Work

Plenty of people counting methods have been proposed recently and almost all people counting methods are based on CNN. In [6] and [7], the problem of people counting has been solved by CNN for the first time. An end to end CNN regression model has been proposed in [6] to finish the task of people counting. The regression model is based on AlexNet network [8] which replaces 4096 neurons of the fully connected layer with a single neuron to count the number of people. In [7], density image has been classified into five levels. In [9], cross-scene counting model has been put forward. The cross-scene counting model can solve the problem that the accuracy of the model will decrease when the scene changes. Different from the above methods that are patch-based training, an end-to-end estimation method has been proposed in [10], where a single image is put into the model to finish the people counting task. Crowdnet has been put forward in [1] which is a combination of deep and shallow convolution network. The Crowdnet can capture semantic information of an image. At the same time, multi-column convolution neural network (MCNN) has been proposed in [2], where three different size filters is used to learn the image features. These three different size filters are designed to capture different level information. Recently, in [3], switching CNN has been proposed, where an optimal regressor can be chose automatically for a particular input patch.

The combination task of different fields has achieved great success, and it has inspired researchers to combine people counting with other tasks. In [11], a model which is based on Resnet-18 [12] has been proposed for people counting, counting density classification and violence detection simultaneously. Differently, a cascaded CNN model, which can estimate density map and classify the people counting into different density levels simultaneously, has been put forward [13]. In [14], density maps that is produced by people counting model has been used to finish the task of counting, tracking and detection. After that, methods to count cars and penguins have been proposed in [15], [16]. The primary contributions of the two papers above are the collection of the cars and penguins datasets. In a higher level cognitive task, a method that combines people counting in calculating the speed of people crossing the road has been proposed in [17].

Among the existing works, there are few effort made on building a model which combines people counting with pedestrian flow statistics. Influenced by the combination task of different fields, we combines people counting with pedestrian flow statistics, which can not only estimate the number of people in each frame of a video, but also count the number of people passing through a place over a period of time.

III. Proposed Algorithm

The proposed algorithm can be divided into two main parts: people counting and pedestrian flow statistics. For the task of people counting, because of the different shooting angles, people who are close to the camera are different from those who are far from the camera. People near the camera usually have high resolution and obvious features, and it is easy for CNN model to learn their head and body features. However, those who are far from the camera have low resolution and blurry features. In order to settle this problem, we propose the CNN model which could simultaneously capture the high and low level features to better represent the people who are far from the camera. For the task of pedestrian flow statistics, density maps generated by CNN model are put into RNN to count the number of people passing through a place over a period of time.

A. People Counting

For the density images, we need to learn high and low level features simultaneously. Therefore we adopt a combination of deep and shallow fully convolution networks to learn the features of density images. The overview of proposed CNN model is shown in Fig. 2.

1) High Level Features Extraction: We use deep fully convolution network to extract high level features. The deep fully convolution network is based on VGGNet-16, which is composed of five group convolution layers and three fully connected layers. Although the VGGNet-16 is originally designed for image classification, it has been used in many computer vision fields such as object detection, object tracking

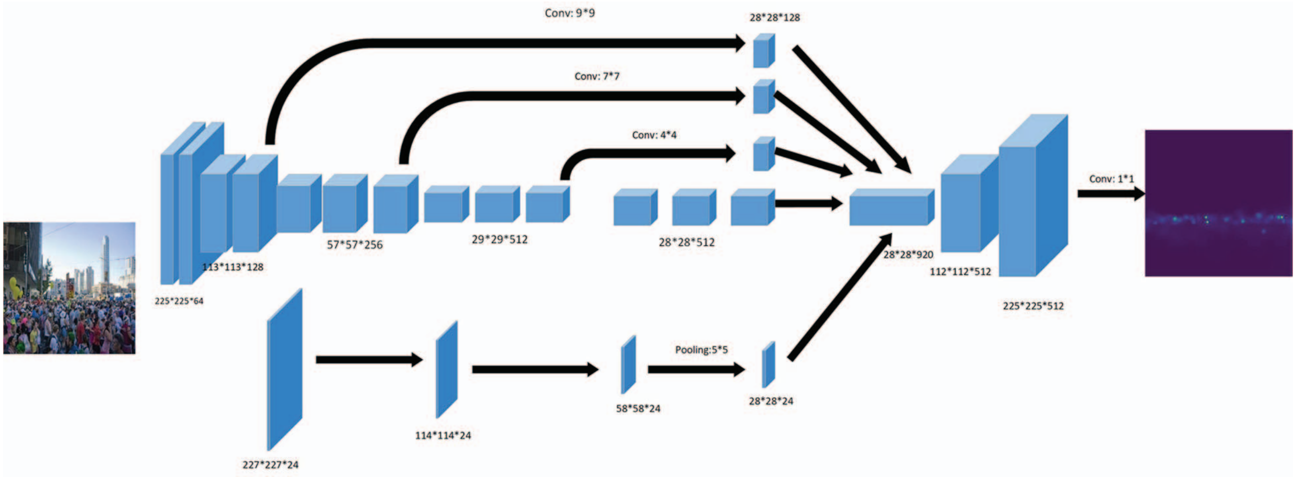

Fig. 2. The overall model of the proposed people counting algorithm.

and object segmentation. Since VGGNet-16 is an excellent generic visual descriptor, we apply it in the task of people counting. However, people counting is different from image classification. Image classification is to classify an entire image while people counting requires pixel-level prediction. Therefore we remove three fully connected layers of VGGNet-16 to get the pixel-level prediction.

As is known to all, VGGNet-16 has a max-pooling layer after each group convolution layer. The stride of max-pooling layer is 2, so the resultant output features have a spatial resolution of 1/32 times the input images. In our works, we need the resultant output features to accomplish the task of pedestrian flow statistics, therefore the resultant output features of large size is required. In order to get larger resultant output features, we set the stride of the fourth max-pooling layer to 1 and remove the fifth max-pooling layer. This can make the resultant output features have a spatial resolution of 1/8 times the input images. Besides, we use the technique of holes [18] to handle the problem of receptive field caused by the change of the stride of the fourth max-pooling layer.

2) Low Level Features Extraction: People who are far from the camera usually only have a head or a leg in images. To recognize these low resolution objects, we adopt a shallow fully convolution network that only have three convolution layers. Each convolution layer has 24 filters, and the kernel size of each filter is 5×5. We add a pooling layer after each convolution layer to make the resultant outputs have the same resolution with resultant outputs of the deep fully convolution network. The shallow fully convolution network is mainly used to detect the small head blobs of density images. Rather than using max-pooling layers, we use the average-pooling layers to ensure that there is no loss of count.

3) Networks Combination: As the resultant outputs of the deep and shallow fully convolution networks have the same resolution, we can concatenate the two resultant output features. Firstly, we extract the Conv2_2, Conv3_3 and Conv4_3 features of the deep fully convolution network to better represent the density images. The features of Conv2_2 layer are processed by a convolution layer with 128 filters, and the kernel size of the each filter is 9×9 with a stride of 4. The features of Conv3_3 layer are processed by a convolution layer with 128 filters, and the kernel size of the each filter is 7×7 with a stride of 2. The features of Conv4_3 layer are processed by a convolution layer with 128 filters, and the kernel size of the each filter is 4×4 with a stride of 1. Secondly, we extract the features of the last layer of the shallow fully convolution network. Thirdly, we concatenate the features which are extracted in deep and shallow fully convolution network respectively. Lastly, we add two deconvolution layers after the concatenating features to make the resultant output features have the same resolution with the input images. The front deconvolution layer has 512 filters, and the kernel size of the each filter is 4×4 with a stride of 4, and the back deconvolution layer also has 512 filters, but the kernel size of the each filter is 3×3.

In total, there are 512 resultant output features, which are then processed by using a 1×1 convolution layer to gain a single output image which is also called density map. Such a density map has the same resolution as the input image.

B. Pedestrian Flow statistics

For the task of pedestrian flow statistics, we need to count the number of people that passing through a place over a period of time. Since the images are successive, the problem of people flow statistics can not be worked out by using CNN alone. Considering that we need to process sequential multimedia data, RNN, especially LSTM [19], can be a better choice. The main contributions of LSTM are the memory cell, input gate, forget gate and output gate. The LSTM also has the ability to maintain its state over time. We have intelligently combined the CNN with LSTM to better estimate the number of people passing through a place over a period of time. The proposed model is shown in Fig. 3. First, density maps are extracted by CNN. Second, a two-layer LSTM is used to learn

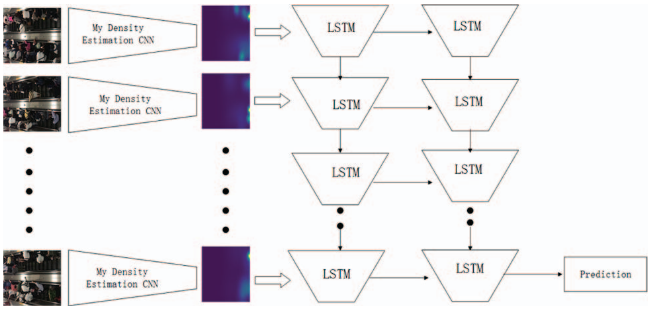

Fig. 3. The pedestrian flow statistics model which combines the people counting model with LSTM.

sequence information of the density maps to count the number of people passing through a place over a period of time.

1) Recurrent Neural Network: RNN is proposed to analyze sequential information in big sequential data, such as video and language. However RNN has the problem of forgetting the earlier inputs of the sequence in case of long term sequences which is called vanishing gradient problem. The problem discussed above can be solved by a special RNN called LSTM. All the RNN architectures have a chain form of repetitive neural network modules. In the standard RNN, the repeating modules only have a very simple structure such as a tanh layer. LSTM has the same architecture, but the duplicate modules have a very different structure showed in Fig. 4. LSTM has the ability to learn long term dependencies, and the LSTM has the architecture of input gates, output gates, and forget gates. The operations performed in LSTM duplicate modules are performed according to Eq. (1) to Eq. (6).

$$f_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_f), \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2}$$

$$\tilde{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C_t}, \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t * tanh(C_t), \tag{6}$$

σ represents sigmoid function. The x_t represents the input at time t and h_t represents the state at time t. f_t represents the forget gate at time t, and f_t decides the previous frame need to be cleared or not. b_f and W_t represent the parameters to be learned. i_t is the input gate which decides whether to write to cell state while b_i and W_i represent the parameters to be learned. Meanwhile, $\tilde{C_t}$ is the recurrent unit which is computed from the input of the current frame x_t and state of the previous frame h_{t-1}. W_c and b_c represent the parameters to be learned. C_t represents the cell state. o_t represents the output gate to maintain the information for the next step. W_o and b_o represent the parameters to be learned. In the end, h_t is the output of the LSTM which indicates the number of people passing through a place.

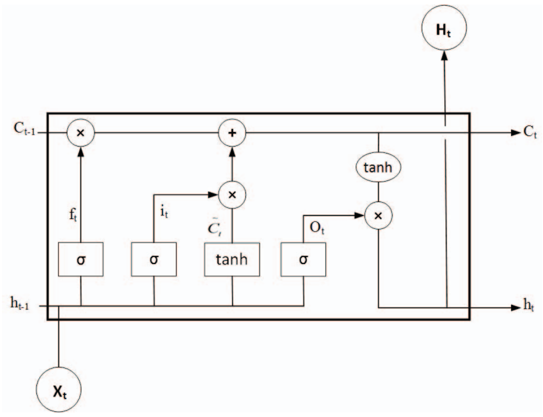

Fig. 4. The structure of LSTM.

2) Two Layers LSTM: The standard RNN only has one single layer, but when considering time sequence information, we need to process the information in more layers. Therefore we stack two LSTM layers, and the outputs of the first layer are the inputs of the second layer. In this way, our model could handle the sequence information more effectively in the task of pedestrian flow statistics.

IV. Experimental Evaluation

In this section, we will evaluate our approach on two challenging datasets including UCF_CC_50 [5] and DH302. For the task of people counting, we use Caffe [20] to extract the density maps and get the caffemodels. After that, PyTorch is used to combine our CNN model with LSTM. The computer configuration is shown as follows: Intel i7-7700, 3.60 GHz CPU with 32 GB RAM and a NVIDIA GeForce GTX 1080 GPU. Similar to the exiting people counting works, we use mean absolute error (MAE) [9] and mean squared error (MSE) [9] to evaluate our model. The MAE and MSE are defined as follows:

$$MAE = \frac{1}{N_{test}} \sum_{1}^{N_{test}} |z_i - \tilde{z}_i|, \tag{7}$$

$$MSE = \sqrt{\frac{1}{N_{test}} \sum_{1}^{N_{test}} (z_i - \tilde{z}_i)^2}, \tag{8}$$

where N_{test} is the number of test images, z_i is the real number of people in the ith image and $\tilde{z}_i$ indicates the estimated number of people in the ith image. The experiments in the two challenging datasets are as follows.

A. UCF_CC_50

UCF_CC_50 is introduced in [5]. The UCF_CC_50 dataset only has 50 gray images with head annotations, but the images have a very big crowd density. The total number of labelled individuals in the entire dataset is 63075 and the average number of people in each image is 1280. The number of people in each image varies from 94 to 4543. The dataset contains

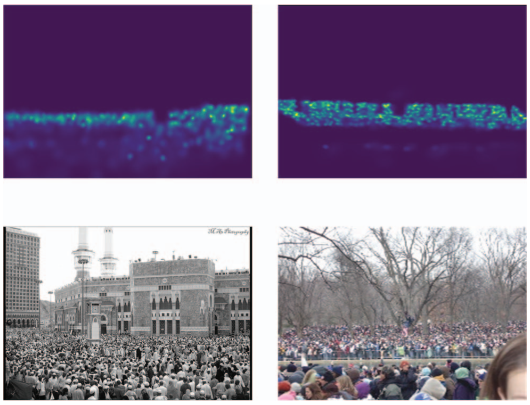

Fig. 5. Original density images and the corresponding density maps.

lots of different scenes such as stadiums, concerts and political rallies.

The density maps in our research are created by using Gaussian kernel normalization which blurs each head annotation. In Fig. 5, we have shown two density maps generated by the Gaussian kernel normalization with the original images. This blurring can not only make the summation over the density maps the same as the total number of people in the images, but also indicate the relationship between the location of the region and the number of people.

UCF_CC_50 have only 50 images, which is not enough to fulfill the requirement of training of CNN. Therefore, we adopt a series methods to augment the UCF_CC_50 dataset. Firstly, we increase the dataset by extracting the images pro-rata 0.5 to 1.2 times of the original images resolution. Secondly, we crop patches from the multi-scale increased images. The patch resolution is 225 × 225 with 50 overlap from the multi-scale increased images. Similar to the previous works, we use a 5-fold cross validation [5] to evaluate our CNN model. We divide the dataset into five splits, and each split contains 10 images. In each fold cross validation, we use four splits to train the CNN model and one split is used to evaluate the performance of the CNN model. We augment the four splits as the augmentation methods described above. Stochastic gradient descent (SGD) optimization is used to train the CNN model, and we set the learning rate to be 0.0000001, weight decay to be 0.005 and momentum to be 0.9.

Table I shows the results of our CNN model along with other recently works on UCF_CC_50. In [21], density maps have been adopt to better finish the task of people counting in density images. In [9], CNN has been firstly used to estimate the density maps, and dense SIFT features have been employed in density images [22]. Compared with these three methods, our method gets better achievement in the dataset of UCF_CC_50.

UCF_CC_50 dataset has only 50 discrete images, which can not be used to finish the task of pedestrian flow statistics. Therefore we have collected two datasets whose name are DH302img and DH302vid to finish the task of pedestrian flow statistics and people counting.

TABLE I
THE RESULTS OF DIFFERENT METHODS ON UCF_CC_50.

Method	MAE	MSE
Rodriguez *et al.* [21]	655.7	697
Zhang *et al.* [9]	467.0	498.5
Lempitsky *et al.* [22]	493.4	487.1
Our Model	445.1	496.4

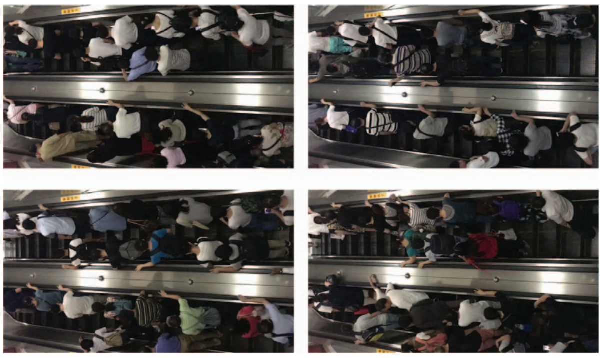

Fig. 6. Examples of the DH302.

B. DH302

We have collected a video whose name is DH302vid in Shanghai metro during rush hours. This video lasts for two hours and has 30 frames per second. For the task of people counting, we randomly select lots of images called DH302img from the DH302vid. The DH302img contains 644 images in which the people is annotated in the center of their heads and contains a total of 10304 people. The resolution of the collected images are 1280 × 720. The maximal number of people in DH302img is 48 while the minimal number is 12. Some of the images are shown in Fig. 6. 512 images of DH302img are used for training data and the rest are used for testing data. For the task of pedestrian flow statistics, we annotate the DH302vid every 10 seconds to indicate the number of passing people and the average number of passing people is 22.

For the task of people counting, DH302img is augmented by using the same methods applied in UCF_CC_50. We also use the 5-fold cross validation to evaluate our CNN model. Table II shows the results of our method along with Crowdnet [1], kernel ridge regression [23] and Gaussian process regression [24] in DH302img. Kernel ridge regression and Gaussian process regression are traditional people counting methods. Crowdnet [1] combines deep and shallow fully convolution network to better extract image information. The results shown in Table II which indicate that our method achieves the best performance in DH302img.

For the task of pedestrian flow statistics, we specially adopt DH302vid which contains 80 minutes training data and 18 minutes testing data. Our video is 30 frames per second

TABLE II
THE RESULTS OF DIFFERENT METHODS ON DH302IMG.

Method	MAE	MSE
Kernel Ridge Regression [23]	8.14	12.34
Gaussian Process Regression [24]	7.81	14.36
Crowdnet [1]	6.14	8.65
Our Model	5.31	8.32

TABLE III
THE RESULTS OF DIFFERENTS METHOD ON DH302VID.

Method	MAE	MSE
LSTM [19]	5.28	6.96
Two-layer LSTM	4.88	6.02
Our Model	4.53	5.81

and we extract 2 images every second in order to reduce computation complexity. Therefore the training data contains 9600 images and the testing data contains 2160 images. The maximal number of people passing through a place over the last 10 seconds is 43 while the minimal number is 14. Table III shows the results of our proposed approach along with the LSTM and the two-layer LSTM. The results indicate that our method achieves the best performance among the three approaches.

V. CONCLUSION

In this paper, people counting is combined with pedestrian flow statistics. By combining people counting with pedestrian flow statistics, we can not only estimate the number of people in each frame of a video, but also count the number of people passing through a place over a period of time. A dataset whose name is DH302img with 664 images has been collected to evaluate the performance of the proposed CNN model. And our CNN model outperforms CrowdNet and MCNN in the datasets of DH302img and UCF_CC_50. A dataset whose name is DH302vid with 11760 images has been collected to evaluate the performance of the proposed pedestrian flow statistics model. Our pedestrian flow statistics model which combines CNN with the LSTM performs better than the model which just uses LSTM in DH302vid.

REFERENCES

[1] L. Boominathan, S. S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *ACM International Conference on Multimedia*, pp. 640–644, 2016.

[2] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 589–597, 2016.

[3] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4031–4039, 2017.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2547–2554, 2013.

[6] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *ACM International Conference on Multimedia*, pp. 1299–1302, 2015.

[7] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, "Fast crowd density estimation with convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 43, pp. 81–88, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, pp. 1097–1105, 2012.

[9] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 833–841, 2015.

[10] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *IEEE International Conference on Image Processing*, pp. 1215–1219, 2016.

[11] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, p. 7, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[13] V. A. Sindagi and V. M. Patel, "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, p. 6, 2017.

[14] D. Kang, Z. Ma, and A. B. Chan, "Beyond counting: Comparisons of density maps for crowd analysis tasks - counting, detection, and tracking," *arXiv preprint arXiv:1705.10118*, 2017.

[15] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *European Conference on Computer Vision*, pp. 483–498, 2016.

[16] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *European Conference on Computer Vision*, pp. 785–800, 2016.

[17] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *European Conference on Computer Vision*, pp. 712–726, 2016.

[18] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *Computer Science*, no. 4, pp. 357–361, 2014.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 2013.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Acm International Conference on Multimedia*, pp. 675–678, 2014.

[21] M. Rodriguez, I. Laptev, J. Sivic, and J. Y. Audibert, "Density-aware person detection and tracking in crowds," in *International Conference on Computer Vision*, pp. 2423–2430, 2011.

[22] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *International Conference on Neural Information Processing Systems*, pp. 1324–1332, 2010.

[23] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," in *IEEE Conference onComputer Vision and Pattern Recognition*, pp. 1–7, 2007.

[24] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *IEEE Conference onComputer Vision and Pattern Recognition*, pp. 1–7, 2008.