A CNN-RNN Neural Network Join Long Short-Term Memory For Crowd Counting and Density Estimation

Jingnan Fu, Hongbo Yang, Ping Liu, Yuzhen Hu

Beijing Information Science and Technology University
No. 12, Xiaoying road, Qinghe, Haidian district, Beijing
Beijing, China
fujingnan612@163.com, anonbo@bistu.edu.cn, spring_cute_lp@sina.com, yz.hu@foxmail.com

Abstract

Crowd counting and density estimation is a challenging task in the field of computer vision. Most of existing methods of this task are based on convolutional neural network (CNN), which have achieved good results in low-density scene. Usually, people who are far away from the camera appear to be denser and smaller, while those who are close to the camera are more sparse and larger, therefor, structure contains only CNN gives the poor performance in some high-density crowd scene because of the uneven distribution of the crowd through camera. To address this problem, this paper designs a CNN-RNN Crowd Counting Neural Network (CRCCNN), which introduces Long Short-Term Memory (LSTM) structure, we use CNN structure to extract the features of the whole image, and use the LSTM structure to extract the contextual information of crowd region. Since LSTM has a good memory of the input information of sequential samples, it can predict the crowd density very well even for the high density population. We perform our experiments on different datasets and compare with other existing methods, which achieve the outstanding results and demonstrate the effectiveness performance of CRCCNN.

Key words: crowd counting, density estimation, CNN, LSTM

Introduction

The influence of the variational crowd density on the safety of many public occasions has attracted more and more attention in recent years. In some sensitive areas, such as subway stations, performance halls, scenic spots, etc. overcrowding in such scenes is easy to cause dicey accidents. Therefore, the estimation of crowd density and real-time supervision of the crowd counting has become an important application of computer vision technology in the field of public security. However, estimating crowd density and counting people accurately in complex scene is still a challenging task.

In some cases where crowd density is extremely dense, as in Fig. 1, is almost impossible to estimate the density and count the people by hand. In recent years, the development of deep learning has brought a great progress to the task of crowd counting and density estimation. Lots of CNN-based algorithms for this task have been proposed. Wang *et al.* [1] who first apply CNNs for the task of crowd density estimation. They proposed an end-to-end deep CNN regression model for counting people from input images, they

Fig. 1 The samples of extremely dense of crowd images.

adopted AlexNet [2] network in their architecture where the final fully connected layer is replaced with a single neuron layer for the purpose of predicting the crowd count. To improve the network's robustness to adapt to different scene datasets for the task of prediction, Zhang *et al*. [3] proposed a CNN based framework for cross-scene crowd counting which is to learn a mapping from images to crowd counts. Recently Zhang *et al*. [4] proposed a novel framework called Multi-column Convolutional Neural Network (MCNN) which contains three columns with different filter size so that the features could be learned by each column CNN is adaptive to variation in people/head size. The method of CNN-based network usually ignores the semantic information in the context of the image. Usually, the crowd close to the camera appear to be sparse, while the crowd far away from the camera are more denser, therefor applying single CNN makes it difficult to estimate density accurately. To overcome this challenge, we design a spacial framework (CRCCNN) which consist of not only CNN but RNN. To adapt to the learning of contextual relevance in crowd images, we adopt LSTM unit in CRCCNN. Our CRCCNN is mainly based on Zhang *et al.* [4], which is improved under the MCNN, for MCNN is a fully convolutional structure, it can be regarded as the process of feature extraction except for the last convolution layer which is for feature fusion. Therefore, the trained MCNN network could extract the crowd features from the whole image perfectly, which is very important for the training of L-

ISBN: 978-1-5386-5609-9

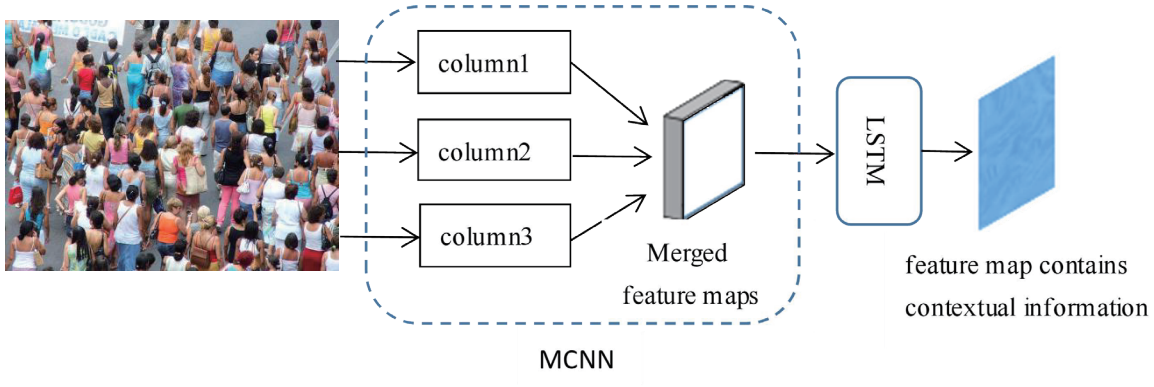

Fig. 2 **The overview of our CRCNN network.** For each of input image, our network firstly computes its features through CNN, then the feature maps will be encoded and as the input of the LSTM unit.

STM. Our CRCCNN allowed input of a whole image and output of a crowd feature map via MCNN, then we use each feature map as the input of LSTM, and learn the contextual relevance inside each feature map. This method can not only preserve the crowd features but also preserve the contextual information of the image.

Related work

In our network, the CNN part completely adopt the MCNN structure, the MCNN parameters are not modified, we replace the loss layer into the LSTM structure, which is mainly follows the reasons: 1) the CNN part is only used to extract the features of the crowd without training, greatly reducing the calculation of the parameter; 2) the design of the multi-columns network can provide better quality map for LSTM training.

LSTM unit in CRCCNN. Thanks to the capability of LSTM in modelling the long short term relationships between sequence elements, recently the combination of CNN and LSTM has been successfully applied in generating image descriptions [5] and video description [6]. CNN-LSTM structure is also used in pedestrian attribute recognition tasks by Wang *et al.* [7]. Inspired by [7], we use the slice layer to decompose the each feature map into 16 regions, each of which is mapped into 1-D vector. After the output of the LSTM unit, all of 16 vectors are spliced and restored into a feature map, as shown in Fig. 3, which is convenient for loss optimization. Being differ from video description task, our task is to segment inside one image, and treat these patches as 16 consecutive frames to learn the regulation of the changeable crowd density. As our model is trained in Caffe, we use the number 16 so as to meet the requirements of input and output blobs of Caffe-LSTM layer.

Images preprocessing. We standardize the images to a normative size rather than cropping or resize them. We stipulate a normative size, in this paper, using 640 × 400 as the size, and for each training image $\boldsymbol{I}$, assumed the shape of $\boldsymbol{I}$ is $h \times w$, the proportionality coeffcients are $S1$ and $S2$ ($S1 = 640 / h$ and $S2 = 400 / w$), then we multiply h and w simultaneously by the minimum between $S1$ and $S2$, therefor we will get the scaled image. The scaled image will be finally centered into a matrix with the size of 640 × 400, larger than

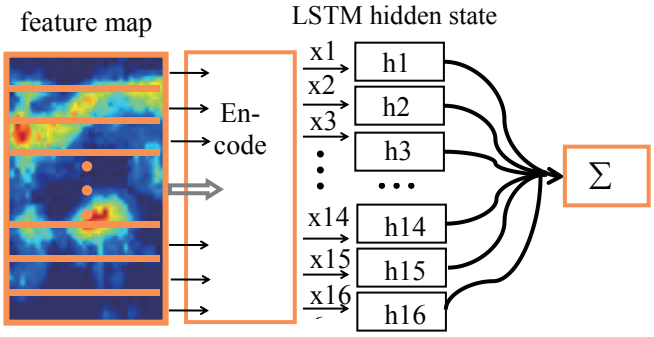

Fig. 3 Structure of LSTM unit in CRCCNN.

the portion of the scaled image will be filled with zero. Thus, except for the region of crowd, the rest region of the image is treated as background. This work can not only avoid distortion but normalize the input images to the same size.

Ground truth. The ground truth contains everyone's location which was labeled by the people head. Every location is defined by abscissa and ordinate. In order to enable the CRCCNN to complete the learning of regression task, we convert the ground truth to density map via geometry-adaptive kernels [4].

Training

A. Data layer

To satisfy the ordered sequences input of LSTM, it is necessary to add a mark to the top of the data layer to hint the beginning and termination of the sequences. In the LSTM unit, this mark is called *cont*, which contains only the values of 0 and 1, represent the end of the previous sequences and the beginning of the next sequences, and the successive sequences, respectively. Therefore, we designed a special python data layer, the top of which contains three blobs: data, label and cont. For each top of data, cont has the shape of 16 × 1, except that the first value is 0, the rest of ones are 1 (*i.e.*, 0, 1, ... , 1), these values will be input to the LSTM unit together with the 16 sequences correspondingly. Since we save the standard density maps as CSV form during the data preprocessing, so we also restore labels to the density maps in

ISBN: 978-1-5386-5609-9

the training stage, each of which is a 2-D array with values ranging from 0 to 1. Each training image (*i.e.*, 16 sequence patches) corresponds to the same label.

B. Model training

As the very limited training samples, it will cause not only the difficulty of gradient convergence but also the large calculation of parameters if we trained the whole CRCCNN directly. Motivated by the successful pre-training of RBM [8], we use the trained weights of MCNN to initialize our CNN structure in all columns and only fine-tune the LSTM unit structure.

C. Optimization of CRCCNN

For our CRCCNN regression networks, we simply adopt the Euclidean distance for loss function. As in function (1), N is the number of training samples, X_i is input of images and θ is the learnable parameters. I is defined as the estimated density maps generated by CRCCNN while I_i is density maps generated by ground truth. Noting that the existence of LSTM structure, our CRCCNN model usually has a large gradient in the initial stage of training, to prevent the gradient from exploding, we introduce a threshold in the training so as to ensure that the gradient descent will eventually be compressed in the threshold to optimize, in this paper, we set the threshold to 5.

$$L(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \left\| I(X_i, \Theta) - I_i \right\|_2^2 . \quad (1)$$

D. Evaluation metric

For evaluating performance of crowd counting and density estimation based on our CRCCNN, we evaluate different methods following MAE and MSE, which are also applied in work [3], the MAE and MSE are defined as follows:

$$MAE = \frac{1}{N}\sum_{1}^{N}\left| z_i - \hat{z}_i \right| \quad , \quad MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}\left(z_i - \hat{z}_i \right)^2} \quad . \quad (2)$$

where N is the number of test samples, z_i is the true number of people in ith image while $\hat{z}_i$ stands for the estimated number of people in ith image. In other words, MAE is equal to the accuracy of results and MSE reflects the robustness of the model.

Experiments

We train our model in Shanghaitech dataset which contains Part_A and Part_B. Our experiments on Shanghaitech dataset consists of two courses, one is to train Part_A and Part_B datasets, respectively, the other is to merge and shuffle Part_A and Part_B (simply called STM dataset) and train on them. For fair comparison, our model is also trained on UCSD datasets. We test CRCCNN on Part_A, Part_B, and UCSD dataset, respectively, and compared the results with different methods.

Results. Table 1 shows the results of different methods on Shanghaitech dataset. Table 2 shows different models perform on UCSD. We also show the results of different models which are trained based on STM dataset in Table 3.

TABLE 1
COMPARRING DIFFERENT METHODS ON SHANGHAITECH DATASETS

Method	Part_A		Part_B	
	MAE	MSE	MAE	MSE
LBP+RR	303.2	371.0	59.1	81.7
Zhang *et al*. [3]	181.8	277.7	32.0	49.8
MCNN	110.2	173.2	26.4	41.3
CRCCNN	**107.0**	**162.1**	**24.3**	**37.8**

As can be seen from Table 1, whether it is training on Part_A or Part_B, the trained CRCCNN model performs better than other methods.

TABLE 2
COMPARRING DIFFERENT METHODS ON UCSD DATASETS

Method	UCSD	
	MAE	MSE
Cumulative Attribute Regression [9]	2.07	6.86
Zhang *et al*. [3]	1.60	3.31
MCNN	1.07	1.35
CRCCNN	**1.04**	1.35

Noting that CRCCNN does not perform better on MSE than MCNN in Table 2. Considering the very sparse crowd density in each image of the UCSD dataset, we can conclude that CRCCNN is better adapt to high-density scenarios.

TABLE 3
COMPARRING DIFFERENT METHODS ON STM DATASETS

Method	STM	
	MAE	MSE
Zhang *et al*. [3]	32.2	57.9
MCNN	28.4	50.1
CRCCNN	**23.1**	**42.4**

Since Part_A and Part_B are dataset of different density levels, it leads to the crowd density of the images changes dramatically in STM dataset, therefor STM is very challenging dataset for testing by different models. As in Table 3, we compared the recent three methods, it is demonstrate that the CRCCNN model in crossover scenarios where the robustness perform better than the other approach. In fact, training CRCCNN networks on the dataset which contains different density level images usually attain good result.

ISBN: 978-1-5386-5609-9

Conclusion

In this paper, we design a CNN-RNN neural network join Long Short-Term Memory (LSTM) for crowd counting and density estimation. We introduce the LSTM structure since we take advantage of the capacity of learning the contextual relevance of sequence elements of LSTM. Usually, crowd images contain a lot of contextual information, it is very difficult for CNN to accurately estimate the crowd count and density of high-density level because of the neglect of contextual information. We test and compare with different methods on different datasets, it is demonstrate that our idea has improved the performance of the prediction results. In addition, We also designed a data layer for the learning task of the multi-sequence inside single image, so that we can preprocess the original data more conveniently. In a word, this paper verify that the task of crowd counting and density estimation needs to consider not only the feature extraction of crowd but also the contextual relevance of crowd density, and we accomplish this task well through our model which based on CNN join LSTM.

References

[1] Wang, C., Zhang, H., Yang, L., Liu, S., Cao, X. Deep people counting in extremely dense crowds, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM. 2015, pp. 1299‑1302.

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097‑1105.

[3] Zhang, C., Li, H., Wang, X., Yang, X. Cross-scene crowd counting via deep convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 833‑841.

[4] Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y., 2016b. Singleimage crowd counting via multi-column convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 589 – 597.

[5] Karpathy A, Li F F. Deep visual-semantic alignments for generating image descriptions. in: IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, pp: 664-676.

[6] Donahue J, Hendricks L A, Rohrbach M, et al. Long-term Recurrent Convolutional Networks for Visual Recognition and Description, in: IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, pp. 677-691.

[7] Wang J, Zhu X, Gong S, et al. Attribute Recognition by Joint Recurrent Learning of Context and Correlation. in: IEEE International Conference on Computer Vision. IEEE Computer Society, 2017, pp. 531-540.

[8] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. NEURAL COMPUT, 2006, pp. 1527-1554.

[9] K. Chen, S. Gong, T. Xiang, and C. C. Loy. Cumulative attribute space for age and crowd density estimation. In CVPR, IEEE, 2013, pp. 2467 – 2474.

ISBN: 978-1-5386-5609-9