Single-Image Crowd Counting via Multi-Column Convolutional Neural Network

Liangjie Cao
10 June 2018

1. Multi-column CNN for density map estimation

Due to perspective distortion, the images usually contain heads of very different sizes, hence filters with receptive fields of the same size are unlikely to capture characteristics of crowd density at different scales. Therefore, it is more natural to use filters with different sizes of local receptive field to learn the map from the raw pixels to the density maps. Motivated by the success of Multi-column Deep Neural Networks (MDNNs) [1], they propose to use a Multi-column CNN (MCNN) to learn the target density maps. In their MCNN, for each column, they use the filters of different sizes to model the density maps corresponding to heads of different scales. For instance, filters with larger receptive fields are more useful for modeling the density maps corresponding to larger heads.

The overall structure of our MCNN is illustrated in Figure 1. It contains three parallel CNNs whose filters are with local receptive fields of different sizes. For simplification, they use the same network structures for all columns (*i.e.*, convCpoolingCconvCpooling) except for the sizes and numbers of filters. Max pooling is applied for each 2x2 region, and Rectified linear unit (ReLU) is adopted as the activation function because of its good performance for CNNs [4]. To reduce the computational complexity (the number of parameters to be optimized), they use less number of filters for CNNs with larger filters. They stack the output feature maps of all CNNs and map them to a density map. To map the features maps to the density map, they adopt filters whose sizes are 1x1 [3]. Then Euclidean distance is used to measure the difference between the estimated density map and ground truth. The loss function is defined as follows:

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \parallel F(X_i; \theta) - F_i \parallel_2^2 \qquad (1)$$

where θ is a set of learnable parameters in the MCNN. N is the number of training image. X_i is the input image and F_i is the ground truth density map of image X_i. $F(X_i; \theta)$ stands for the estimated density map generated by MCNN which is parameterized with θ for sample X_i. L is the loss between estimated density map and the ground truth density map.

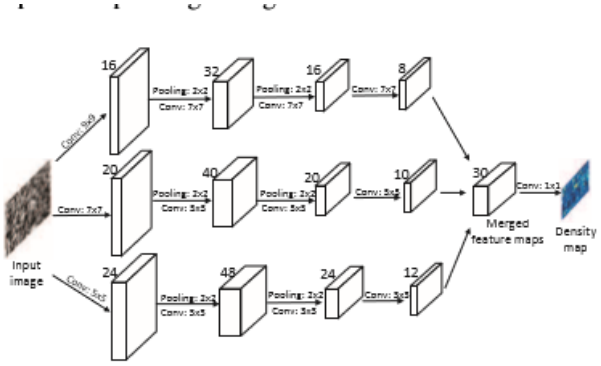

Figure 1. **The structure of the proposed multi-column convolutional neural network for crowd density map estimation**

2. Optimization of MCNN

The loss function 1 can be optimized via batch-based stochastic gradient descent and backpropagation, typical for training neural networks. However, in reality, as the number of training samples are very limited, and the effect of gradient vanishing for deep neural networks, it is not easy to learn all the parameters simultaneously. Motivated by the success of pre-training of RBM [2], they pre-train CNN in each single column separately by directly mapping the outputs of the fourth convolutional layer to the density map. They then use these pre-trained CNNs to initialize CNNs in all columns and fine-tune all the parameters simultaneously.

3. Transfer learning setting

One advantage of such a MCNN model for density estimation is that the filters are learned to model the density maps of heads with different sizes. Thus if the model is trained on a large dataset which contains heads of very different sizes, then the model can be easily adapted (or transferred) to another dataset whose crowd heads are of some particular sizes. If the target domain only contains a few

Dataset	Resolution	Num	Max	Min	Ave	Total
UCSD	158x238	2000	46	11	24.9	49885
UCF_CC_50	different	50	4543	94	1279.5	63794
WorldExpo	576x720	3980	253	1	50.2	199923
Shanghaitec A	different	482	3139	33	501.4	241677
Shanghaitec B	768x1024	716	578	9	123.6	88488

Table 1. **Figure/ground benchmark results**

training samples, we may simply fix the first several layers in each column in our MCNN, and only fine-tune the last few convolutional layers. There are two advantages for fine-tuning the last few layers in this case. Firstly, by fixing the first several layers, the knowledge learnt in the source domain can be preserved, and by fine-tuning the last few layers, the models can be adapted to the target domain. So the knowledge in both source domain and target domain can be integrated and help improve the accuracy. Secondly, comparing with fine-tuning the whole network, fine-tuning the last few layers greatly reduces the computational complexity.

4. Evaluation metric

By following the convention of existing works [5] for crowd counting, we evaluate different methods with both the absolute error (MAE) and the mean squared error (MSE), which are defined as follows:

$$MAE = \frac{1}{N}\sum_{1}^{N} \mid z_i - \bar{z_i} \mid, MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(z_i - \bar{z_i})^2} \tag{2}$$

where N is the number of set test images, z_i is the actual number of people in the ith image, and $\bar{z_i}$ is the estimated number of people in the ith image. Roughly speaking, MAE indicates the accuracy of estimates, and MSE indicates the robustness of estimates.

5. Shanghaitech dataset

As exiting datasets are not entirely suitable for evaluation of the crowd count task considered in this work, they introduce a new large-scale crowd counting dataset named Shanghaitech which contains 1198 annotated images, with a total of 330,165 people with centers of their heads annotated. As far as we know, this dataset is the largest one in terms of the number of annotated people. This dataset consists of two parts: there are 482 images in Part A which are randomly crawled from the Internet, and 716 images in Part B which are taken from the busy streets of metropolitan areas in Shanghai. The crowd density varies significantly between the two subsets, making accurate estimation of the crowd more challenging than most existing datasets. Both Part A and Part B are divided into training and testing: 300 images of Part A are used for training and the remaining 182 images for testing, and 400 images of Part B are for training and 316 for testing. Table 1 gives the statistics of Shanghaitech dataset and its comparison with other datasets. They also give the crowd histograms of images in this dataset in Figure 2. If the work is accepted for publication, they will release the dataset, the annotations, as well as the training/testing protocol.

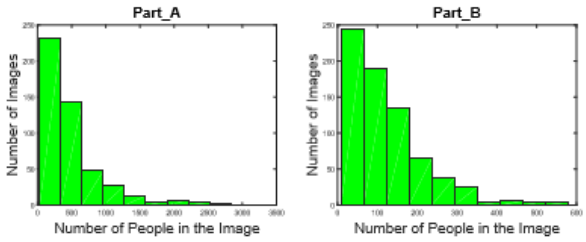

Figure 2. **Histograms of crowd counts of our new dataset**

References

[1] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012.

[2] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *NEUTAL COMPUT*, 2006.

[3] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[4] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. W. Senior, V. Vanhoucke, J. Dean, et al. On rectified linear units for speech processing. In *ICASSP*, 2013.

[5] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015.