

Article

A High-Density Crowd Counting Method Based on Convolutional Feature Fusion

Hongling Luo ^{1,2}, Jun Sang ^{1,2,*} , Weiqun Wu ^{1,2}, Hong Xiang ^{1,2}, Zhili Xiang ^{1,2}, Qian Zhang ^{1,2} and Zhongyuan Wu ^{1,2}

¹ Key Laboratory of Dependable Service Computing in Cyber Physical Society of Ministry of Education, Chongqing University, Chongqing 40004, China; hollyluo@cqu.edu.cn (H.L.); wuwq@cqu.edu.cn (W.W.); xianghong@cqu.edu.cn (H.X.); zhili.xiang@cqu.edu.cn (Z.X.); zhngqn@cqu.edu.cn (Q.Z.); zhongyuandt@gmail.com (Z.W.)

² School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China

* Correspondence: jsang@cqu.edu.cn; Tel.: +86-139-8369-7592

Received: 20 October 2018; Accepted: 20 November 2018; Published: 23 November 2018



Abstract: In recent years, the trampling events due to overcrowding have occurred frequently, which leads to the demand for crowd counting under a high-density environment. At present, there are few studies on monitoring crowds in a large-scale crowded environment, while there exists technology drawbacks and a lack of mature systems. Aiming to solve the crowd counting problem with high-density under complex environments, a feature fusion-based deep convolutional neural network method FF-CNN (Feature Fusion of Convolutional Neural Network) was proposed in this paper. The proposed FF-CNN mapped the crowd image to its crowd density map, and then obtained the head count by integration. The geometry adaptive kernels were adopted to generate high-quality density maps which were used as ground truths for network training. The deconvolution technique was used to achieve the fusion of high-level and low-level features to get richer features, and two loss functions, i.e., density map loss and absolute count loss, were used for joint optimization. In order to increase the sample diversity, the original images were cropped with a random cropping method for each iteration. The experimental results of FF-CNN on the ShanghaiTech public dataset showed that the fusion of low-level and high-level features can extract richer features to improve the precision of density map estimation, and further improve the accuracy of crowd counting.

Keywords: Convolutional Neural Network (CNN); feature fusion; crowd counting; density map

1. Introduction

Crowd safety in public places has always been a significant but troublesome affair, especially in high-density crowd gathering places. The higher the crowd density is, the easier of out of control will be [1], and it can cause serious casualties. It is important to seek an intelligent method of crowd analysis in public places to assist in prevention and decision making. As an important part of crowd analysis [2], crowd counting and density estimation can help to quantify the importance of events and provide relevant personnel with information to support decision-making. Therefore, crowd counting and its density estimation become hot topics in the security field, which are widely used in video surveillance, traffic monitoring, public safety and urban planning [3]. Nowadays, there is a very large demand for a crowd monitoring system. However, the existing crowd monitoring system products still have many weakness, for example, most of them are limited by application scenes or have low precision. Especially, research on monitoring the number of pedestrians in a large-scale crowded environment (Refer to Figure 1) is insufficient [4].



Figure 1. Examples of crowded scenes.

The crowd counting methods can be divided into the detection-based methods and the regression-based methods. The detection-based crowd counting methods usually use a sliding window to detect each pedestrian in the scene to obtain the approximate position of the pedestrian, and then count the number of pedestrians [5–7]. The detection-based methods can achieve good results for low-density crowd scenes, but are greatly limited for high-density crowd scenes. The incipient regression-based methods try to learn the direct mapping between the low-level features extracted from the local image blocks and the head count [8–10]. Such direct regression-based methods only count the number of pedestrians, while ignoring the important spatial information. Refs. [11,12] suggested learning the linear or non-linear mapping between the local block features and their corresponding target density maps, which may incorporate the spatial information into the learning process.

The great success of the Convolutional Neural Network (CNN) in many computer vision tasks motivated researchers to use CNN to learn nonlinear functions from crowd images to corresponding density maps or corresponding counts. In 2015, Wang et al. [13] applied CNN to the crowd counting task firstly by using the Alexnet network structure [14]. The fully connected layer with 4096 neurons was replaced by a layer with only one neuron to count the number of pedestrians in the crowd image. In the same year, Zhang et al. [4] noticed that the performance of the existing methods was drastically reduced when they were applied to new scenes that differed from the training dataset. To overcome this problem, a data-driven method was proposed to fine-tune the pre-trained CNN model with the training samples similar to the density level in the new scene to adapt to the unknown application scenes. This method avoids the problem of re-training when the model is transformed to a new scene, but still needs a large amount of training data, and it is difficult to know the density level of the new scene in advance in practical applications. Inspired by the success of multi-column networks [15] in image recognition, in 2016, Zhang et al. [16] proposed a multi-column convolutional neural network-based architecture (MCNN) by constructing a network consisting of three columns of filters corresponding to the receptive fields with different sizes (large, medium, small) to adapt to changes of head size due to the perspective effects or image resolution. During training, each column of the MCNN pre-trains all image blocks, then combines the three networks for fine-tuning training. The training process is complex and there is a large amount of structural redundancy. In 2017, Sam et al. [17] proposed switching the convolutional neural network for crowd counting (Switching CNN) to train the regressions by using a particular set of training data patches according to different

crowd densities in the image. The network consists of multiple independent CNN regressions, which is similar to a multi-column network, while a Switch classifier based on the VGG-16 [18] architecture was introduced to select the optimal regression for a particular input block. The Switch classifier and the independent regression are trained alternately. However, Switching CNN uses the Switch classifier to switch between the regressions, which is very expensive and often inaccurate. Similar to Refs. [16,17], in 2017, Kumaga et al. [19] considered that a single predictor in different scene environments is not sufficient to predict the number of pedestrian properly, and thus proposed a hybrid neural network Mixture of CNNs (MoCNN). The model structure consists of a mixture of expert CNNs and a gated CNN. The appropriate expert CNN is selected adaptively upon the background of the input image. In prediction, expert CNNs estimate the head count in the image, while gated CNN predicts the appropriate probability of each expert CNN. These probabilities are further used as weighting factors to calculate a weighted average of the head count estimated by all expert CNNs. MoCNN not only trains multiple expert CNNs, but also learns the probability of the head count estimated by each expert CNN through a gated CNN training. However, it can only be used for crowd counting estimation, while it cannot get the crowd density distribution information. Tang et al. [20] proposed a low-rank and sparse-based deep-fusion convolutional neural network for crowd counting (LFCNN) by adopting a regression method based on low-rank and sparse penalty to promote the accuracy of the projection from the density map to global counting, which got an excellent performance. Zhang et al. [21] proposed scale-adaptive CNN (SaCNN) to estimate the crowd density map and integrate the density map to get a more accurate estimated head count by extracting feature maps from multiple layers and adapted them to have the same output size. Han et al. [22] combined convolutional neural network and Markov Random Field (CNN-MRF) to achieve the head count in static images which contained three parts: a pre-trained deep residual network 152 [23] to extract features, a fully connected neural network for count regress and a MRF to smooth the counting results of the local patches. In this way, high correlation of local adjacent patches was used to improve count accuracy.

For the crowd counting task under high-density and complex environments, a feature fusion-based deep convolutional neural network method, FF-CNN (Feature Fusion of Convolutional Neural Network), was proposed in this paper to obtain more accurate crowd counting performance. FF-CNN aimed to map the crowd image to its crowd density map, and then obtained the head count by integration. Following MCNN [16], the geometry-adaptive kernels were adopted to generate high-quality density maps which were used as the training ground truths. VGG network was used as the trunk network of FF-CNN to obtain richer features. Deconvolution technique [24,25] was used to achieve the fusion of high-level and low-level features. Two loss functions i.e., density map loss and absolute count loss, were used to jointly optimize for obtaining a more precise density map and a more accurate crowd count. In order to increase the sample diversity, the original images were cropped to images of 256×256 with a random cropping method for each iteration.

2. Feature Fusion CNN for Crowd Counting

2.1. Density Map Based Crowd Counting

There are two options for estimating the head count of a crowd image with CNN: (1) Input the crowd image and output the estimated head count directly; (2) input the crowd image, and then output the crowd density map which shows the head count of each pixel in the image, and then integrate any area in the density map to get a final head count in the area. Compared with outputting head count directly, the estimated crowd density map may show the spatial distribution information of a given image. In addition, when learning density maps with a CNN model, the learned filters are more suitable for different sizes of human heads, which can improve the robustness to perspective changes. Thus, these filters have more semantic information and can improve the accuracy of crowd counting [16]. Therefore, we will learn the nonlinear mapping between the crowd image and the corresponding density map with CNN, and get the head count by integrating the density map.

2.2. Ground Truth Density Map

To apply supervised training on FF-CNN by means of density map regression from an input image, the image with labeled pedestrian heads should be converted into a ground truth density map. The ground truth density map generation method was used in FF-CNN [14].

Suppose a pedestrian head is at pixel x_i , which can be represented as a delta function $\delta(x - x_i)$. For N labeled pedestrian heads in an image, it can be expressed as follows:

$$H(x) = \sum_{i=1}^N \delta(x - x_i) \quad (1)$$

Convert it into a continuous density function with Gaussian kernel G_σ [7]:

$$F(x) = \sum_{i=1}^N \delta(x - x_i) \times G_\sigma(x) \quad (2)$$

where N is the head count, σ is the variance of Gaussian kernel. The sum of the density maps is equivalent to the total number of pedestrians.

Considering the perspective transformation, it is necessary to obtain the Gaussian spread parameter σ according to the size of each pedestrian head in the image. However, in a high-density environment, it is almost impossible to accurately obtain the size of the occluded head manually. It is observed that the distance between two adjacent pedestrian heads in a crowded scene is proportional to the size of the pedestrian head. Therefore, a geometrically adaptive Gaussian kernel can be used to generate a density map [16].

Suppose that the crowd is distributed evenly around each pedestrian head, the average distance between each pedestrian head and its k neighbors may provide a viable geometric distortion estimation method. For a given head coordinate x_i , suppose its distance from k neighbors is $\{d_1^i, d_2^i, \dots, d_k^i\}$, the average distance is $\bar{d}^i = \frac{1}{k} \sum_{j=1}^k d_j^i$ and set β as the initial variance. The density function is:

$$F(x) = \sum_{i=1}^N \delta(x - x_i) \times G_{\sigma_i}(x), \sigma_i = \beta \bar{d}^i \quad (3)$$

Although we used the adaptive density map generation method of Ref. [16], the value of β was optimized to obtain the ground truth density map with higher quality, which will be introduced in Section 3.2.

2.3. Network Architecture

The VGG [18] network has achieved outstanding results in the image classification competition, which has strong expansion and generalization performance. The network model used in this paper was based on the VGG-16 network, and the backbone of the network was constructed by repeatedly stacking the convolutional layer with 3×3 convolutional kernel and the max pooling layer with 2×2 filter. Since too much downsampling will result in smaller feature maps and lower resolutions, to obtain a higher-quality crowd density map, only the first four sets of convolutions in the VGG-16 network will be adopted, which includes three downsampling layers. The first four sets of convolutions in the VGG-16 network are shown in Figure 2:

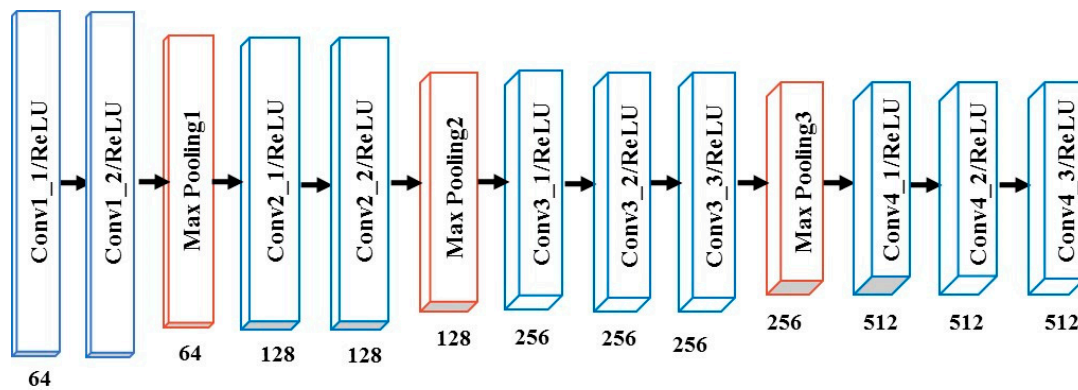


Figure 2. The first four sets of convolutions in the VGG-16 network.

The experiments with CCF (Convolutional Channel Features) [26] demonstrated that, to extract richer features, the suggested typical accumulated downsampling coefficients are four and eight. Inspired by this conclusion, the corresponding feature maps of downsampling coefficients of four and eight (the feature maps of Conv3_3 and Conv4_3) were fused by deconvolution technique to get richer features and finally get higher-quality crowd density map.

The constructed network architecture in this paper is shown in Figure 3. The convolutional layers from Conv1_1 to Conv4_3 use a small convolutional kernel of 3×3 . The convolutional step is fixed to 1 pixel, and a pixel padding with a value of 0 is performed around the input space of the convolutional layer, which may keep spatial resolution after convolution operation. A rectified linear unit (ReLU) [27] is applied to each convolutional layer as an activation function. All pooling layers use the max pooling operation, while the filter size is 2×2 with a fixed step size of 1. The number of feature maps of the convolutional layer will be doubled after each downsampling. After twice max pooling operations with filter size 2×2 , the size of the feature map Conv3_3 is $W/4 \times H/4$, and Conv4_3 has undergone three times downsampling operations to obtain a feature map of size $W/8 \times H/8$. In order to achieve the fusion of Conv3_3 and Conv4_3 feature maps, a deconvolution layer is added after the Conv4_3 to sample the feature maps of Conv4_3 to the same size as the feature maps of Conv3_3. Conv5_1 and Conv5_2 of the convolution layer with 3×3 filters are adopted in the upper layer of the network to reduce the number of feature maps and increase the nonlinearity of the model. Finally, a convolutional layer with 1×1 filters is used to map it to the density map, which can be integrated to obtain the overall count.

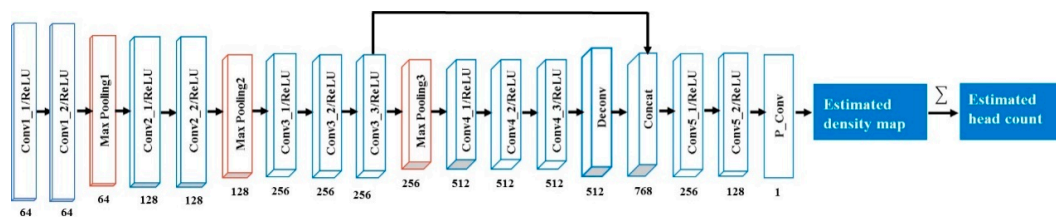


Figure 3. FF-CNN network architecture. FF-CNN: Feature Fusion of Convolutional Neural Network.

Since a deconvolution layer after the Conv4_3 layer is added, the size of the feature map of the Conv4_3 layer is sampled to the size of the feature map of Conv3_3 layer. Therefore, though there are three downsampling operations in the network, the final accumulated downsampling coefficient is four. It only needs to downsample the ground truth density map to $1/4$ of the input image, while the resulted estimated density map size is also $1/4$ of the original image. In addition, though FF-CNN is a fully convolutional network [28,29], which may accept images with any sizes, since there is a concatenate layer in the network, the output sizes of Conv3_3 and Conv4_3 layers should be identical after convolutional and pooling layers, so the height and width of the input image should be a multiple of eight.

2.4. Network Loss

The network training uses Euclidean loss to measure the distance between the estimated density map and the ground truth density map:

$$L_D(\theta) = \frac{1}{M} \sum_{i=1}^M \|F_d(X_i; \theta) - D_i\|^2 \quad (4)$$

where θ is the set of parameters to be learned in the network, M is the total number of training images, X_i is the input image, and D_i is the corresponding true density map. $F_d(X_i; \theta) - D_i$ represents the density map estimated for X_i . The Euclidean distance is applied to each pixel, and then accumulated. The purpose of Equation (4) is to obtain a high quality crowd density distribution to finally obtain an accurate crowd count.

Since the estimated crowd count is expected to be as close as possible to the ground truth head count after training, in addition to density map loss function, another count loss function is also introduced. The absolute loss function with weight was denoted as:

$$L_y(\theta) = \frac{1}{M} \sum_{i=1}^M \|F_y(X_i; \theta) - Y_i\|^2 \quad (5)$$

where $F_y(X_i; \theta)$ and Y_i represent the estimated head count and the ground truth head count, respectively.

3. Experiments

ShanghaiTech dataset was used to evaluate our proposed FF-CNN model. Implementation of the proposed network and its training were based on the Caffe framework provided by the Berkeley Center for Vision and Learning (BCVL). The computer used for experiments was with Intel (R) Xeon CPU E5-2683 v3 @ 2.00Ghz, while the GPU was NVIDIA TESLA K80 (NVIDIA, Santa Clara, CA, USA). The experimental platform was equipped with 64-bit ubuntu14.04, Anaconda3.4, CUDA Toolkit8.0 and Opencv2.7.0.

3.1. Dataset

The experiments were conducted on the challenging ShanghaiTech dataset [16], which not only has different density levels, but also has different complex scenes, such as different scales and different perspective distortions. It can be seen in Table 1, the ShanghaiTech dataset consists of two parts: Part A and Part B, including a total of 1198 images and 330,165 labeled heads. Part A includes 482 images randomly selected from the Internet. The images in Part B were taken from street photographs in Shanghai. Compared to Part B, Part A includes the images with higher density. For experiments, both of these two parts were divided into training and test sets. Three-hundred and 182 images in Part A were used for training and testing, respectively, while 400 and 316 images in Part B were used for training and testing, respectively.

Table 1. ShanghaiTech dataset. Num is the number of images; Max is the maximum crowd count; Min is the minimal crowd count; Total is total number of labeled pedestrian.

Dataset	Scenes	Num	Resolution	Min	Max	Total
Part A	different	482	different	33	3139	241,677
Part B	different	716	1024 × 768	9	578	88,488

3.2. Implementation Details

In our experiments, different from Ref. [14], which set the value of the parameter β in Equation (3) as $\beta = 0.3$, we set $\beta = 0.12$, which showed the best result. Also, same as Ref. [16], the size of the head was limited within 100 pixels, i.e., when $\bar{d}^i > 100$, let $\bar{d}^i = 100$ ($\bar{d}^i$ is a parameter in Equation (3)). Other parameters were set following Ref. [16].

During training, all parameters were optimized with Batch gradient descent (BGD) and Back Propagation (BP). The initial learning rate was 0.00001, the momentum was 0.9, and the weight decay was set to be 0.0001 to avoid over-fitting during training. The loss weight of the density map was set to be 1 and the absolute count loss weight was 0.00004. The density map loss function and the absolute count loss function were trained together. The total number of iterations was set to 2,200,000 (batch size was set to 1) both on Part A and Part B datasets. For Part A dataset, 300 iterations were regarded as one epoch and the total training epoch was 7333. For Part B dataset, 400 iterations were regarded as one epoch and the total training epoch was 5500.

For the original training dataset, a random crop method to crop nine patches from each image was used to augment training images and each patch was 1/4 size of the original image in Ref. [16]. In order to further speed up training, while reducing the memory usage and enhancing the diversity of the data, the training data set images were cropped randomly in each training iteration. Since the height and width of the input image should be a multiple of eight, the cropped image block size was fixed to 256×256 , while the cropping position was random.

3.3. Evaluation Metrics

The mean absolute error (MAE) and mean square error (MSE) were used to evaluate performance [15–17]. The MAE reflects the accuracy of the prediction and the MSE reflects the robustness of the prediction. The detailed definitions are as follows:

$$MAE = \frac{1}{M} \sum_{i=1}^M |z_i - \hat{z}_i| \quad (6)$$

$$MSE = \sqrt{\frac{1}{M} \sum_{i=1}^M (z_i - \hat{z}_i)^2} \quad (7)$$

where M is the number of test images, z_i and $\hat{z}_i$ represent the ground truth head count and the estimated head count in the i -th image, respectively. With smaller values of MAE and MSE, the results will be better.

3.4. Results and Analysis

The proposed method was compared with other existing methods and the experimental results were shown in Table 2. “LBP + RR” (local binary pattern + Ridge Regression) is a traditional regression-based method, which uses LBP features extracted from the original image as input and uses RR to estimate the head count for each image [4]. It can be seen that the method based on traditional artificial feature extraction is very ineffective for complex high-density crowd datasets, and the CNN-based method is far superior to the traditional method.

Also, it can be concluded from Table 2 that the results of FF-CNN are much better than Cross-scene [4], MCNN [16], FCN [29], Cascaded-MTL [30] and Switching-CNN [17] method. Compared to LFCNN [20] and SaCNN [21], the proposed FF-CNN method shows lower error on the crowd-intensive Part A dataset. MAE and MSE for LFCNN are reduced by 7.45 and 3.1, respectively. MAE and MSE for SaCNN are reduced by 5.05 and 0.4, respectively. The difference is not obvious in the more sparse Part B datasets. MAE and MSE for our proposed method are only 1.75 and 0.79 higher, respectively, than those of LFCNN. MAE and MSE for our proposed method are only 0.25 and 0.39 higher, respectively, than those of SaCNN. However, LFCNN should go through two phases: deep-fusion density map regression and low-rank

and sparse based regression. The FF-CNN proposed needs only one phase and the performance is excellent; SaCNN is more complex than the FF-CNN proposed in this paper. SaCNN needs more training parameters. Also, the resolution of the density map of SaCNN is only 1/8 of the original image, while the resolution of the density map obtained by FF-CNN is 1/4 of the original image. Compared to CNN-MRF [22], the proposed FF-CNN method shows relatively higher error on the crowd-intensive Part A, especially the error of MSE is higher than 8.7 which shows that CNN-MRF is more robust. On the Part B dataset, the proposed FF-CNN method performs better on MAE. The CNN-MRF contains three parts, a pre-trained deep residual network 152 to extract features, a fully connected neural network for count regress and a MRF to smooth the counting, which resulted in more parameters (residual network 152 contains more than 40 million parameters, our method contains about 12 million parameters) and calculations, but not much improvement.

Table 2. Experimental results on ShanghaiTech dataset with different methods.

Method	Part A		Part B	
	MAE	MSE	MAE	MSE
LBP + RR [4]	303.2	371.0	59.1	81.7
Cross-scene [4]	181.8	277.7	32.0	49.8
MCNN [16]	110.2	173.2	26.4	41.3
FCN [29]	126.5	173.5	23.76	33.12
Cascaded-MTL [30]	101.3	152.4	20.0	31.1
Switching-CNN [17]	90.4	135.0	21.6	33.4
LFCNN [20]	89.2	141.9	14.7	25.4
SaCNN [21]	86.8	139.2	16.2	25.8
CNN-MRF [22]	79.1	130.1	17.8	26.0
FF-CNN	81.75	138.80	16.45	26.19

Figure 4 shows the ground truth head count and the estimated head count in the 182 test images of the Part A dataset. Figure 5 shows the ground truth head count and the estimated head count in the 316 test images of the Part B dataset. The dashed line indicates the ground truth head count, while the solid line indicates the head count estimated (the ground truth head count with the corresponding estimated head count is arranged in ascending order to get a more intuitive broken line diagram). It can be seen that either for Part A or Part B, the estimation head count is close to the ground truth head count. Note that for Part A, the crowd density ranges from 66 to 2255, while for Part B, the crowd density ranges from 11 to 539, which shows quite different density and varied density distribution of the crowd. However, the FF-CNN can be effectively adaptive to different density levels and the estimation results are very accurate.

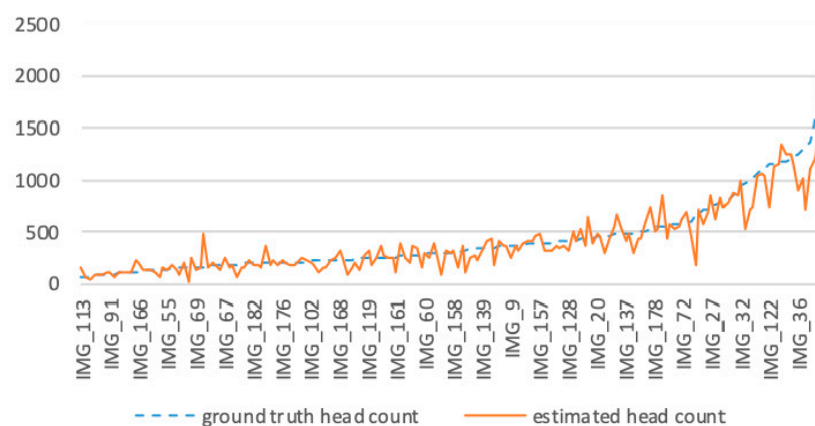


Figure 4. The ground truth head count and estimated head count with FF-CNN on Part A dataset (the vertical axis represents the head counts, and the horizontal axis represents the test sample).

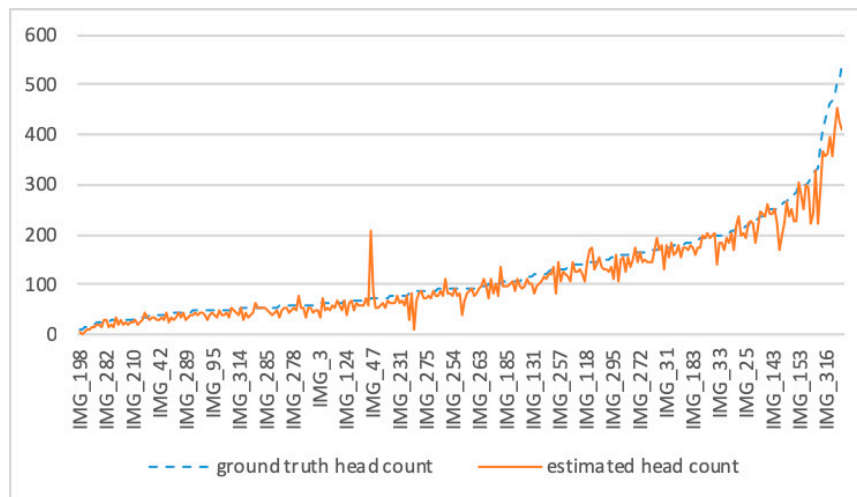


Figure 5. The ground truth head count and estimated head count with FF-CNN on Part B dataset (the vertical axis represents the head counts, and the horizontal axis represents the test sample).

Figure 6 shows several experimental results on the Part A test sets and Part B test sets. The left side shows the test crowd images, while the right side shows the density maps estimated by the proposed model. It can be seen that the density map estimated by the method proposed in this paper can restore the crowd distribution in the image well and adapt to different scenes, light, occlusion, different sizes, and perspective changes effectively. The estimation is very close to the ground truth.

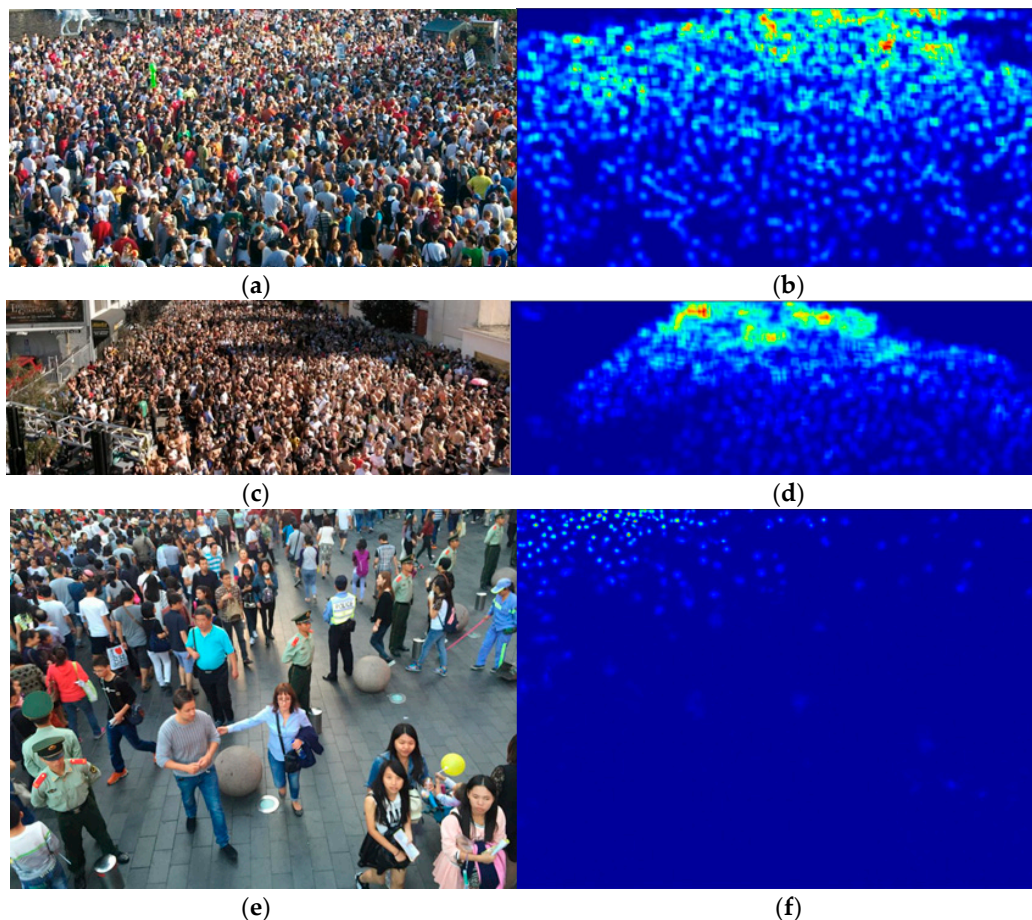


Figure 6. Cont.

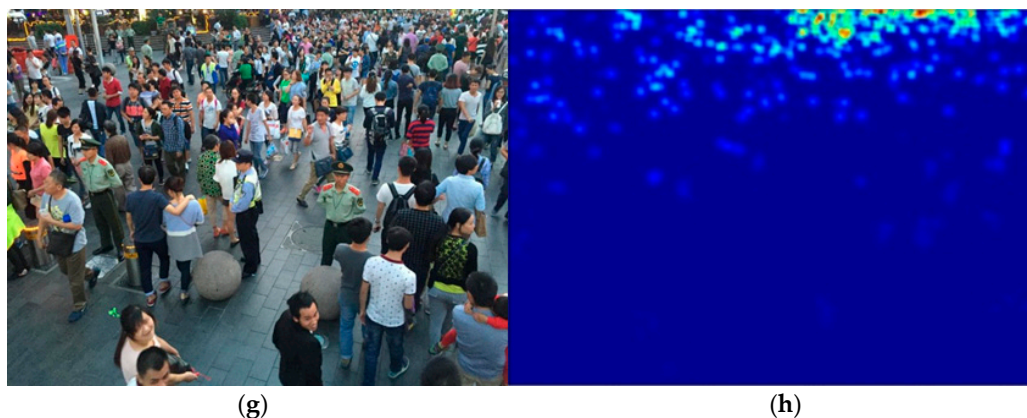


Figure 6. Experimental results on ShanghaiTech test set, (a) IMG_17 on Part A test sets (Ground truth: 1155); (b) estimated density map of (a) (Estimation: 1140.92); (c) IMG_172 on Part A test sets (Ground truth: 1171); (d) estimated density map of (c) (Estimation: 1173.49); (e) IMG_216 on Part B test sets (Ground truth: 161); (f) estimated density map of (e) (Estimation: 156.96); (g) IMG_293 on Part B test sets (Ground truth: 327); (h) estimated density map of (g) (Estimation: 329.34).

4. Conclusions

In this paper, a deep convolutional neural network method based on feature fusion (FF-CNN) was proposed to solve the problem of crowd counting and density distribution estimation in crowded scenes. The proposed FF-CNN mapped the crowd image to its corresponding crowd density map and got the crowd count by integrating the crowd density map. For obtaining a higher-quality crowd density map, the geometry adaptive kernels were adopted to generate high-quality ground truth density maps during training and the deconvolution technique was used to achieve the fusion of high-level and low-level features to get richer features. The absolute count loss with weight 0.00004 was used to get a more accurate crowd count. For enhancing the generalization ability of FF-CNN, the original images were cropped with a random cropping method for each iteration to increase the sample diversity. The experimental results on the ShanghaiTech dataset showed that the fusion of low-level features and high-level features can extract richer features and can get comparable results with state-of-the-art methods. Also, it is adaptive to different scenes and different crowd density levels, and is robust to scale and perspective changes.

Author Contributions: Conceptualization, H.L. and J.S.; Data curation, W.W. and Z.W.; Funding acquisition, H.X.; Investigation, Z.X.; Methodology, H.L. and W.W.; Project administration, J.S.; Software, Z.X. and Q.Z.; Supervision, J.S.; Validation, H.L., H.X. and Z.W.; Visualization, W.W.; Writing—original draft, H.L. and W.W.; Writing—review & editing, J.S., H.X. and Q.Z.

Funding: This work was funded by National Key R&D Program of China (Grant No. 2017YFB0802400).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Fruin, J.J. *Pedestrian Planning and Design*; Metropolitan Association of Urban Designers & Environmental Planners: New York, NY, USA, 1971.
2. Zhan, B.; Monekosso, D.; Remagnino, P.; Velastin, S.; Xu, L.-Q. Crowd analysis: A survey. *Mach. Vis. Appl.* **2008**, *19*, 345–357. [[CrossRef](#)]
3. Zeng, L.; Xu, X.; Cai, B.; Qiu, S.; Zhang, T. Multi-scale convolutional neural networks for crowd counting. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 465–469.

4. 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, Boston, MA, USA, 8–10 June 2015; pp. 833–841.
5. Leibe, B.; Seemann, E.; Schiele, B. Pedestrian detection in crowded scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005; pp. 878–885.
6. Zhao, T.; Nevatia, R.; Wu, B. Segmentation and tracking of multiple humans in crowded environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1198–1211. [[CrossRef](#)] [[PubMed](#)]
7. Ge, W.; Collins, R.T. Marked point processes for crowd counting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2913–2920.
8. Chan, A.B.; Liang, Z.S.J.; Vasconcelos, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008; pp. 1–7.
9. Ryan, D.; Denman, S.; Fookes, C.; Sridharan, S. Crowd counting using multiple local features. In Proceedings of the Digital Image Computing: Techniques and Applications, Melbourne, Australia, 1–3 December 2009; pp. 81–88.
10. Chan, A.B.; Vasconcelos, N. Bayesian poisson regression for crowd counting. In Proceedings of the IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 545–551.
11. Lempitsky, V.; Zisserman, A. Learning to count objects in images. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–9 December 2010; pp. 1324–1332.
12. Pham, V.Q.; Kozakaya, T.; Yamaguchi, O.; Okada, R. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3253–3261.
13. 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, Brisbane, Australia, 26–30 October 2015; pp. 1299–1302.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
15. Schmidhuber, J.; Meier, U.; Ciresan, D. Multi-column deep neural networks for image classification. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3642–3649.
16. Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; Ma, Y. Single-image crowd counting via multi-column convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 589–597.
17. Sam, D.B.; Surya, S.; Babu, R.V. Switching convolutional neural network for crowd counting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5744–5752.
18. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
19. Kumagai, S.; Hotta, K.; Kurita, T. Mixture of Counting CNNs: Adaptive Integration of CNNs Specialized to Specific Appearance for Crowd Counting. *arXiv*, 2017; arXiv:1703.09393.
20. Zhang, L.; Shi, M.; Chen, Q. Crowd counting via scale-adaptive convolutional neural network. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, 12–14 March 2018; pp. 1113–1121.
21. Tang, S.; Pan, Z.; Zhou, X. Low-Rank and Sparse Based Deep-Fusion Convolutional Neural Network for Crowd Counting. *Math. Probl. Eng.* **2017**, *2017*, 1–11. [[CrossRef](#)]
22. Han, K.; Wan, W.; Yao, H.; Hou, L. Image Crowd Counting Using Convolutional Neural Network and Markov Random Field. *arXiv* **2017**, arXiv:1706.03686.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535.

25. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
26. Yang, B.; Yan, J.; Lei, Z.; Li, S.Z. Convolutional channel features. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 82–90.
27. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
28. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 3431–3440.
29. Marsden, M.; McGuinness, K.; Little, S.; O'Connor, N.E. Fully convolutional crowd counting on highly congested scenes. In Proceedings of the 12th International Conference on Computer Vision Theory and Applications, Porto, Portugal, 27 February–1 March 2017; pp. 27–33.
30. Sindagi, V.A.; Patel, V.M. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In Proceedings of the IEEE International Conference on In Advanced Video and Signal Based Surveillance, Lecce, Italy, 29 August–1 September 2017; pp. 1–6.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).