Multimedia Tools and Applications (2020) 79:15059–15071
https://doi.org/10.1007/s11042-020-08888-5

Arbitrary perspective crowd counting via local to global algorithm

Chuanrui Hu[1] · Kai Cheng[1] · Yixiang Xie[1] · Teng Li[1]

Received: 31 March 2019 / Revised: 16 March 2020 / Accepted: 27 March 2020 /
Published online: 18 April 2020

Abstract
Crowd counting is getting more and more attention. More and more collective activities, such as the Olympics Games and the World Expo, are also important to control the crowd number. In this paper, we address the problem of crowd counting in the crowded scene. Our model accurately estimated the count of people in the crowded scene. Firstly, we proposed a novel and simple convolutional neural network, called Global Counting CNN (GCCNN). The GCCNN can learn a mapping, transforms the appearance of image patches to estimated density maps. Secondly, the Local to Global counting CNN (LGCCNN), calculating the density map from local to global. Stiching the local patches constrains the final density map of the larger area, which makes up for the difference values in the perspective map. In general, it makes the final density map more accurate. The dataset we used is a set of public dataset, which are WorldExpo'10 dataset, Shanghaitech dataset, the UCF_CC_50 dataset and the UCSD dataset. The experiments have proved our method achieves the state-of-the-art result over other algorithms.

Keywords Crowd density map · Convolutional neural network · Perspective distortion

1 Introduction

The crowd counting has important social significance and market value. Managers can reasonable scheduling of manpower, material resources and optimize resources configuration by using the number of ROI area statistics. For some of the square, passageway and other public occasions, the result of crowd statistics have very good warning effect to social security problems. Therefore, the crowd counting becomes the key point in the field of video analysis and intelligent video surveillance. This involves estimating the number of people in the crowd and the crowd distribution over the entire region.

Chuanrui Hu and Kai Cheng are contributed equally to this work

✉ Teng Li
liteng@ahu.edu.cn

[1] Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Electrical Engineering and Automation, Anhui University, Hefei, 230601, China

Traditional crowd counting algorithms share a common procedure. 1) Foreground segmentation, but the split of foreground can not entirely separate people and background because sometimes people are still in high density scenarios, such as the queue in front of the station ticket window. 2) Crowd feature extraction, due to the dense scenarios perspective distortion, brightness condition and low resolution of the image, the handcraft features(eg, Scale Invariant Feature Transform(SIFT) [13], Histogram of Oriented Gradient(HOG) [3], Local Binary Patterns(LBP) [16]) cannot fully express characteristics of the crowd. For the dense crowd, typical static crowd scenes come from the WorldExpo'10 Dataset [19].

It is difficult to detect the number of people because of occlusion, and it is not wise to calculate the number by the foreground segmentation due to the randomness of foreground segmentation. Some typical scenes from the WorldExpo'10 dataset [19] are shown in Fig. 1.

In recent years, there has been tremendous progress in certain area [14] of computer vision built on the success of deep learning. There has been a significant recent progress in the field of crow counting due to the development of deep learning (eg the convolutional neural networks (ConvNets)) [12, 19]. To the best of our knowledge, Zhang et al. [2] first train a CNN model to learn a map to solve the crowd counting problem, but in order to get the crowd count, the result need to feed a ridge regressor with the output features. The MCNN [22], which output is an estimated density map and it solve the large scale variation. But the output final estimated density map is distortion due to the size of the the final estimated density map is decreased. Recent researches [19, 22] have proven the learned features performed better than the traditional hand-crafted features. As illustrated in Fig. 2, in order to make up for the shortcoming of the resent search [19, 22], we propose our convolutional neural network architectures to learn the regression function that mapping the image appearance into a crowd density map. The number of people in the crowd scene is calculated through integration over the crowd density map.

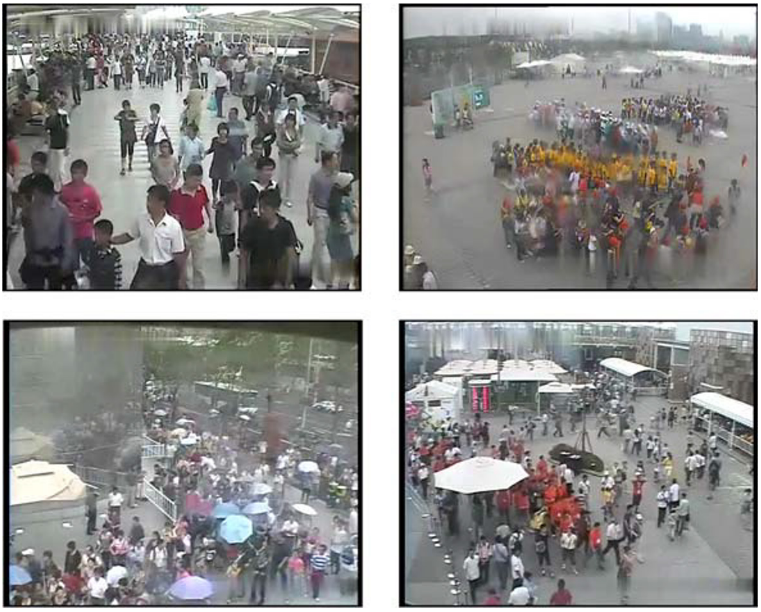


Fig. 1 Sample crowd scene from the WorldExpo'10 dataset

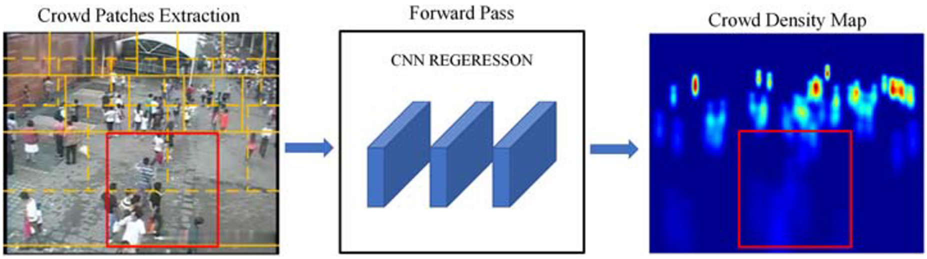


Fig. 2 We define the crowd counting task like a regression problem where a CNN model to map the appearance of image to crowd density map. The yellow box indicates that the training image dataset is densely extracted from the whole image

The main contributions of this work can be conclude into these three aspects.

- In Section 3.1, we propose a novel convolutional neural network architectures, named Global Counting CNN (GCCNN). Which is a fully convolutional neural network [12] can get an accurate regression of a crowd density map of image patches. Since we adopted a bilinear interpolation algorithm, Fig. 4 clearly shows that the final output feature map is the same size as the input patch.
- Due to the scale variation in the crowd images, we introduce the Local to Global Counting CNN (LGCCNN) in Section 3.2 which provide an algorithm, calculating the final density map form local to global. The algorithm make up for the differences caused by different values in the perspective map then makes the density map of the larger area more accurate.
- Our architecture has been evaluated on three benchmark datasets and is shown to achieve state-of-the-art outperforms.

The rest of this paper are organized as follows: previous research about the crowd counting is in Section 2. The proposed method and the overall structure of the two CNN models are detailed listed in Section 3. Experiments and the comparisons of results are summarized in Section 4. In the end, we make a conclusion about this paper in Section 5.

2 Related works

In recent years, the crowd counting method in the literature can be divided into two categories: counting by detection and counting by regression.

Counting by detection [4, 9, 15, 21]. Many algorithms counting people by detection. First, they use the appearance and the motion feature to separate the moving objects from the background over the two consecutive frames of a video clip. Then these algorithms utilize the handcraft features (such as Haar wavlet features or edgelet features [21]) to obtain the moving objects. However, these methods can be used in the video clip not suit for the still image and the handcrafted features often sustain a decline in accuracy when the scene is perspective distortion, severe overlapping, and varying illumination.

Counting by regression [1, 6–8, 11, 14, 18, 19, 22]. Counting by regression aims to lean a mapping between the low-level features and people count via certain a regression function without foreground segmentation or pedestrian detection. It is more suitable for complex environments and more crowded instance like pedestrians. Zhang et al. [2] first trained a deep CNN model. It makes good performance. But they reported the results feeding a ridge regressor with the output features of their CNN model and the input patches of their CNN model is random which does not consider the large scale-invariant to large scale changes well. Our network diminish the perspective distortion and estimates both the crowd count as well as the crowd density map.

3 Methodology

In this section, we will state our notation and crowd counting methodology. Here, we treat the crowd counting problem as the density map estimation.

Previous research has followed [10, 19] and defined the groundtruth of the density map regression as sum of the Gaussian kernels centered on the locations of objects. This mentioned density map is more suitable for representing the density distribution of circle-like objects like cells and bacteria. Considering the shape of the pedestrian in an ordinary surveillance camera is ellipse-like. We follow the method of the [19]. Before generate the groundtruth density map, we should consider the large scale variation due to the perspective distortion. Perspective normalization is necessary to describe the pedestrian scale. After we get the perspective map of each scene and a set of head annotations images, where all the heads are marked by dots. We can generate the groundtruth density map D_i , for an image I, is defined as a sum of Gaussian functions centered on each dot annotation. We generate the crowd density map is generated as:

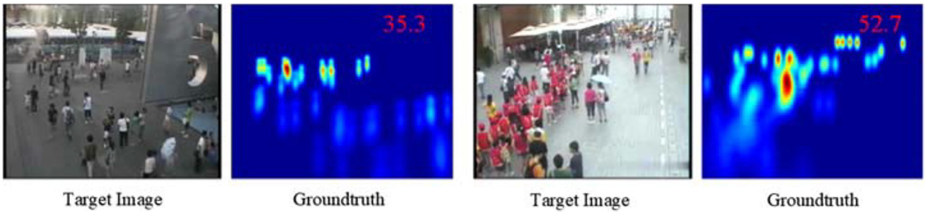
$$D_i(P) = \sum_{P \in P_i} \frac{1}{\|Z\|}(\mathcal{N}_h(P; P_h, \sigma_h) + \mathcal{N}_b(P; P_b, \Sigma)) \tag{1}$$

Where P_i is the set of 2D points of the image I, $\mathcal{N}_h(P; P_h, \sigma_h)$ and $\mathcal{N}_b(P; P_b, \Sigma)$ respectively represent a normalized 2D Gaussian kernel as a head part and a normalized 2D Gaussian kernel as a body part. P_h is the head position and P_b is the body position, estimated by the head position value and the value in the perspective map. Some groundtruth density maps is shown in Fig. 3.

Our CNN model is to learn a non-linear regression function that takes an image patch P with associated groundtruth density map and groundtruth crowd count. As an assistant object, the crowd count associated with the training patch is integrated from groundtruth density map. It returns an estimated density map $D^{(P)}_{pred}$.

$$D^{(P)}_{pred} = F(P|\theta_{net}) \tag{2}$$

Where θ_{net} is the set of parameters of the CNN model. For the image patch P, we could get the $D^{(P)}_{pred}$. Thus for a given unseen test image, at first our algorithm densely extracted



Target Image Groundtruth Target Image Groundtruth

Fig. 3 Crowd images with their corresponding groundtruth density maps

image patches over the image. Then our CNN model could generate an estimated density map corresponding to the image patch. At last, all the density maps are aggregated into a density map for a whole test image.

3.1 The global counting CNN model

Let us introduce our first ConvNet structure called the Global Counting CNN model (GCNN). As illustrated in Fig. 4.

The crowd density estimation does not like image classification, it need per pixel predictions. So we adopt the fully convolutional neural networks natural. This would reduce the overfitting due to the fully convolutional neural network has much fewer parameters than a network trained on an entire image. The structure consists of 6 convolution layers and 2 pooling layers. They are specially designed to extract the crowd features. The Conv1 layer has 3×3 filters with a depth of 64. The Conv2 layer has 3×3 filters with a depth of 128. The max pooling layer with a 2×2 kernel size is used after conv1 and conv2. The Conv1 layer has 33 filters with a depth of 256. The Conv4 layer and Conv5 layer are made of 1×1 filters with a depth of 1000 and 400. The Conv6 layer is another 1×1 filters with a depth of 1. The output from these convolution layers is upsampled to the size of the input image patch using bilinear interpolation to directly obtain the estimated crowd density map.

Due to the good performance for the CNNs of the Parametric Rectified Linear Unit (PReLU) [5], the PReLU was adopted as the activation function and it is not shown in the Fig. 2. Equation (2) has point out, our CNN models is to learn a mapping from a set of features extracted from training image patches to an estimated crowd density map. So, our GCCNN is trained to solve the regression problem. The Euclidean distance is used as the loss function.

$$L1\left(\theta net\right) = \frac{1}{2N}\sum_{i}^{N}||F\left(Pi|\theta net\right) - D_{gt}^{(Pi)}||^2 \tag{3}$$

$$L2\left(\theta net\right) = \frac{1}{2N}\sum_{i}^{N}||C\left(Pi|\theta net\right) - C_{gt}^{(Pi)}||^2 \tag{4}$$

Where θnet denotes the learned parameters of the CNN model, N is the number of the training images, Pi is the image patch will be training in the CNN model. $F(Pi|\theta net)$ and $C(Pi|\theta net)$ represent the corresponding image patch stand for the estimated crowd density map and the crowd count. D_{gt}^{Pi} and C_{gt}^{pi} respectively represent the groundtruth density map and ground truth crowd number of the corresponding image patch. Different from Zhang et al., the master loss task is the $L1(\theta net)$. We let the two loss functions pass through all

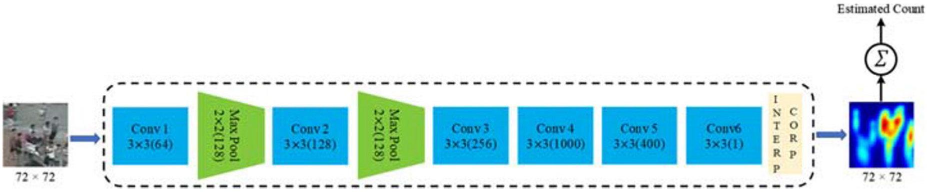


Fig. 4 Our GCCNN structure, treated the input patches and their associated groundtruth density maps and groundtruth crowd counts as input, which returns an estimated density map, the size is same as the input patch

previous layers together. The master loss task is the $L1(\theta net)$. The $L2(\theta net)$ is treated as the auxiliary loss. The auxiliary loss task helps optimize the learning process, while the master loss task takes the most responsibility. We add weight to balance the auxiliary loss. The two loss tasks assisted each other and trained together to obtain optimization.

After obtaining the parameters θnet of the CNN model. How do we implement the prediction stage on the unseen target test image? First, we densely extracted image patches. Then all the image patches are resized to 72×72 pixels. These input image patches with their associated groundtruth density maps and groundtruth crowd count are as illustrated in Fig. 5, which passed through our CNN architecture. It returns an estimated density map corresponding to the input image patch. Lastly, all the output estimated density map will be aggregated into a density map over the whole test image. Due to the extracted image patches are overlap. So the each location of the final estimation density map must by normalized according the number of patches that calculated into the final estimated density map.

3.2 The local to global counting CNN model

On the basis of a counting by regression model, using the annotated perspective map of each scene to solve the perspective distortion and scale variation. Due to the impact of the perspective distortion on each image, the size of pedestrian will exhibit scale variation. The features extracted from the same pedestrian at different scene depths would have notable differences in values.

Go a step further, in order to get an accurate estimated crowd density map, we use the Local to Global Counting CNN model (LGCCNN). We proposed an algorithm for estimating a density map from local to global which is specialized in the perspective distortion and scale variation. The ConvNet structure was specialized designed is shown in Fig. 6.

Our CNN model was consisted of three columns CNN. The three parallel CNNs contain the same structure(i.e., conv-pooling-conv-pooling) and the same size of filters. The CNN model takes different but related inputs. The input is the training image patches cropped from the training images. The patch of the first column was resized to 94×94 pixel. The next two columns take the upper and lower two parts of a complete patch. Each parallel CNN is in charge of learning features of input patch for a different perspective value. Then

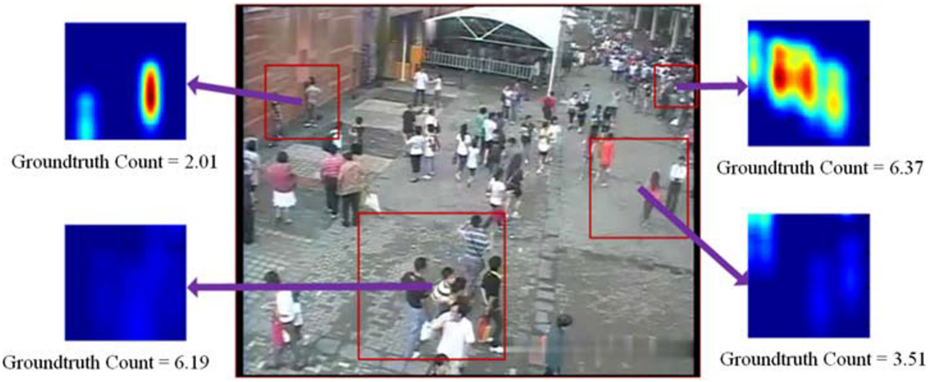


Fig. 5 Patches with their associated labels

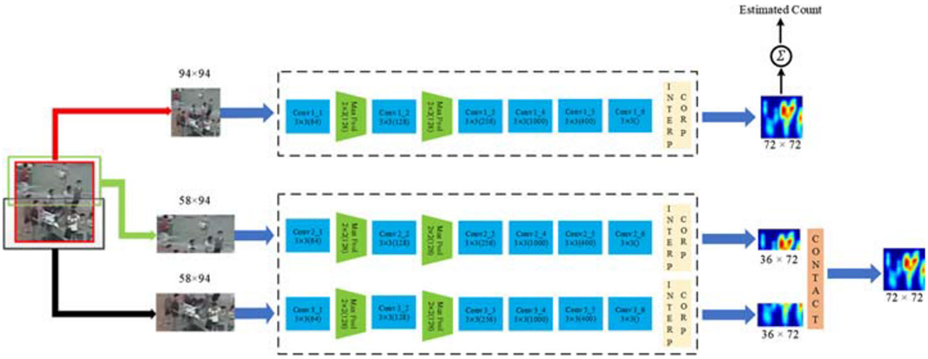


Fig. 6 The CNN architecture for LGCCNN

the output feature maps of the last two columns CNN model are stitched. Compared the losses of the GCCNN, we added the same loss function as show in (7).

$$L1\left(\theta net_1\right) = \frac{1}{2N}\sum_{i}^{N}||F\left(P_i|\theta net_1\right) - D_{gt}^{(P_i)}||^2 \tag{5}$$

$$L2\left(\theta net_1\right) = \frac{1}{2N}\sum_{i}^{N}||C\left(P_i|\theta net_1\right) - C_{gt}^{(P_i)}||^2 \tag{6}$$

$$L3\left(\theta net_2\right) = \frac{1}{2N}\sum_{i}^{N}||F\left(P_{i_1}; P_{i_2}|\theta net_2\right) - F(P_i|\theta net_2)||^2 \tag{7}$$

Where θnet_1 denotes the learned parameters of the LGCNN model in the first column, θnet_2 denotes the learned parameters of the LGCNN model in the next two columns. N is the number of the training images, P_i is the image patch will be training in the CNN model. P_{i_1} and P_{i_2} are the corresponding upper and lower image patch of the completed image patch P_i. $F(Pi|\theta net)$ is the output feature map of the first column CNN model. Noticed, the upper and lower estimated density maps were calculated by the different values of the perspective map. We constrain the final estimated density map on the lager region which makes up for the difference caused by the different perspective values. In the end, it makes the estimated density map more accurate and provide a method for crowd density map was from local to global.

4 Experiments

We first evaluate our CNN model on the challenging the WordExpo'10 dataset [19]. The detail of the WorldExpo'10 dataset is shown in Table 1. This dataset contains 1132 annotated video clips, captured by 108 surveillance cameras. 1,127 one-minute long video sequences are treated as training datasets. Testing datasets, 5 one-hour long different video sequences. Each video sequence contains 120 labeled frames. We train our deep convolution neural network on the basis of caffe library and some modifications are applied. The NVIDIA GTX TITAN X GPU is used. We use the standard Stochastic Gradient Descent(SGD) algorithm to optimize ConvNet parameters with a learning rate of 1e - 3 and momentum of 0.9. The training epoches is 60. During the experiment, we found that if we trained directly on our

Table 1 The attribution of the public datasets: NUM is number of frames; Total is the number of labeled people; MAX is the maximum number of people in the ROI of a frame; MIN is the minimum number of people in the ROI of a frame. AVG indicated the average crowd count

Dataset	NUM	TOTAL	MAX	MIN	AVG
UCSD	2000	49885	46	11	25
UCF_CC_50	50	63974	1279	4543	1279
WorldExpo'10	4.44 million	199623	253	1	50
ShanghaiTech Part A	482	241677	3139	33	501
shanghaiTech PArt B	716	88488	578	9	123

CNN model. The final trained model would have a good effect in certain sences, so we used imagenet dataset training to fine tuning our CNN model and get the final result as shown in Table 2.

4.1 Evaluation criteria

In order to make the experimental results more intuitionistic. We use the two evaluation criteria: the mean absolutely evaluation(MAE) and the mean square evaluation(MSE).which are defined as follows:

$$MAE = \frac{1}{N}\sum_{1}^{N}|C_i - E_i| \tag{8}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}|C_i - E_i|^2} \tag{9}$$

Where N denotes the number of the training images, C_i is the true pedestrians number of the ith test image. E_i is the estimated pedestrians number of ith test image. MAE represents the actual situation of the estimates error. MSE represents the robustness of the estimates. And the smaller the value of MAE and MSE, the more accurate the counting result.

4.2 Data preprocessing

The dataset consists of 108 scenes. In order to train our GCCNN model, we typically selected 2600 images from the 103 scenes in the dataset. We collected 200 patches of 72×72 pixels extracted all over the image with their associated groundtruth density maps

Table 2 Quantitative results with other state of the art methods on the WorldExpo'10 dataset. Only MAE

Method	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Avg
LBP+RR [17]	13.6	58.9	37.1	21.8	23.4	31.0
Zhang et al. [2]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN. [22]	3.4	20.6	12.9	13.0	8.1	11.6
ACSCP [19]	2.8	14.05	9.6	8.1	2.9	7.5
CP-CNN [20]	2.9	14.7	10.5	10.4	5.8	8.9
GCCNN	7.5	22.6	15.7	16.0	6.2	13.6
LGCCNN	2.6	19.3	17.4	14.8	4.7	11.0

and groundtruth crowd counts. It contains 100 positive patches which center is the area of people and the 100 negative patches which center is the area of the ground. Then we performed a data augmentation by flipping each patch randomly.

To train our LGCCNN model, we performed the same method as previously mentioned to extract the image patches of 94×94 pixels. For a given complete patch of 94×94 pixels, we will get the upper and lower patch of 58×94 pixels by cropped the complete patch.

We typically selected 2600 images from the 103 scenes in the dataset as the training images. Firstly, we collected 200 patches of 94×94 pixels extracted all over the images with their associated groundtruth density maps and groundtruth crowd counts . It contains

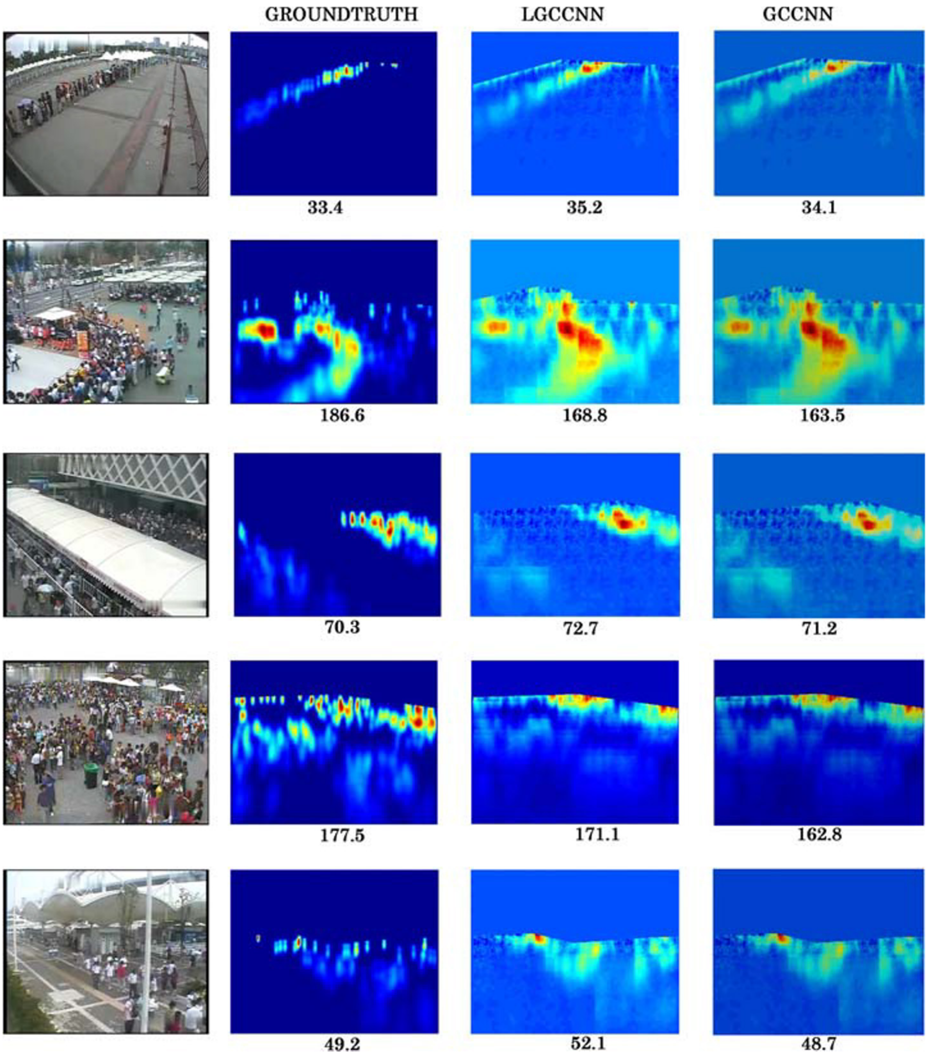


Fig. 7 Sample predictions of our LGCCNN model and GCCNN model in the World-Expo’10 dataset. The first column is the target test image. The second column is the groundtruth density map corresponding to the target test image. The third an fourth columns show the estimated density maps for LGCCNN model and GCCNN model, respectively

Table 3 Comparing performances of different methods on Shanghaitech dataset, the UCF_CC_50 dataset and the UCSD dataset

Method	The Shanghaitech dataset				The UCF_CC_50		The UCSD	
	PartA		PartB		dataset		dataset	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
LBP+RR [17]	303.2	371.0	59.0	81.7	—	—	—	—
Zhang et al. [2]	181.8	277.7	32.0	49.8	467.0	498.5	1.60	3.31
MCNN [22]	110.2	173.2	26.4	41.3	337.6	509.1	1.07	1.35
ACSCP [19]	75.7	102.7	17.2	27.4	291.0	404.6	1.04	1.35
CP-CNN [20]	73.6	106.4	20.1	30.1	295.8	320.9	-	-
LGCCNN	105.2	169.8	25.6	40.3	336.5	510.2	1.05	1.27

100 positive patches which center is the area of people and the 100 negative patches which center is the area of the ground. Then we peform a data augmentation by flipping each patch randomly. After we got these patches and in order to meet our CNN model. We will collect the small local patches by cropped the global patches.

4.3 Results

Our GCCNN data preprocessing is similar to the method of Zhang et al. [2]. In the Table 2, our GCCNN model gets a better performance in scene 1 and scene 5. These two subdatasets contain pedestrians is about 80 in each image which is more suitable in the actual world. By contrast, the best performance is LGCCNN, which is reduced the MAE effectively. The LGCCNN combined the local and the global features, is well to solve the problem of perspective distortion and scale variation. The ACSCP and CP-CNN solve the problem of crowd counting from different aspects, generative adversarial networks and crowd density level. Using the generative adversarial networks which will lead to poor convergence in the training procedure. The CP-CNN fusions crowd density level feature and image feature to gain a more accurate density map, which need extra information. However, we proposed a LGCCNN to deal with crowd counting, considering calculating the density map from local to global. Stiching the local patches constrains the final density map of the larger area, which make up for the difference values in the perspective map. Extensive experimental results show that the proposed method is effective and it does not need extra information and complex training procedure. Figure 7 shows some of qualitative results of the WorldExpo' 10 dataset that are obtained by the LGCCNN model and GCCNN model. Table 3 shows the experimental results that our algorithm with some algorithms on the Shanghaitech dataset, the UCF_CC_50 dataset and UCSD dataset.

5 Conclusions

In this paper, we proposed two convolution neural network architectures. For our first architecture, the GCCNN model can learn a mapping which transforms the appearance of crowd image to the crowd density map effectively. Our second architecture, the LGCCNN model which goes a step further, provide a method that was from local to global for crowd density map. The final estimated density map on the lager region which makes up for the difference caused by the upper and lower image patch of different perspective values. In the end, it makes the estimated density map more accurate. The density map is generated in the output layer of network and the number of people is obtained by integral regression. We test our proposed method in the Shanghaitech dataset, the WorldExpo'10 dataset, the UCF_CC_50 dataset and the UCSD dataset. Moreover, the experimental results show the accuracy the robustness of our method outperforms the state-of-the-art crowd counting method.

Acknowledgements This work is supported by the National Key R&D Program of China (No. 2018YFB1305804), and the Anhui Provincial Natural Science Foundation of China (No. 1908085J25).

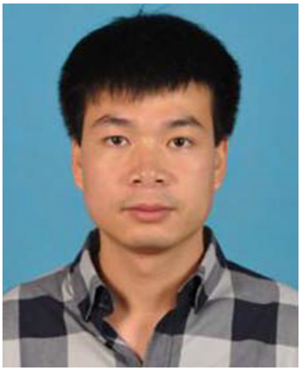
References

1. Chan AB, Vasconcelos N (2012) Counting people with low-level features and bayesian regression. IEEE Trans Image Process 21(4):2160–2177
2. Cong Z, Li H, Wang X, Yang X (2015) Cross-scene crowd counting via deep convolutional neural networks. In: IEEE conference on computer vision & pattern recognition
3. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Schmid C, Soatto S, Tomasi C (eds) International conference on computer vision & pattern recognition (CVPR '05), vol 1. IEEE Computer Society, San Diego, pp 886–893. https://doi.org/10.1109/CVPR.2005.177. https://hal.inria.fr/inria-00548512
4. Ge W, Collins RT (2009) Marked point processes for crowd counting. In: IEEE conference on computer vision & pattern recognition
5. He K, Zhang X, Ren S, Jian S (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification
6. Hu Y, Chang H, Nian F, Yan W, Teng L (2016) Dense crowd counting from still images with convolutional neural networks. J Vis Commun Image Represent 38(C):530–539
7. Ke C, Chen CL, Gong S, Tao X (2012) Feature mining for localised crowd counting. In: British machine vision conference
8. Lempitsky VS, Zisserman A (2010) Learning to count objects in images. In: International conference on neural information processing systems
9. Lin SF, Chen JY, Chao HX (2001) Estimation of number of people in crowded scenes using perspective transformation. Systems Man & Cybernetics Part A Systems & Humans IEEE Transactions on 31(6):645–654
10. Liu R, Chen Y, Zhu X, Hou K (2016) Image classification using label constrained sparse coding. Multimed Tools Appl 75(23):15619–15633
11. Liu T, Tao D (2014) On the robustness and generalization of cauchy regression. In: IEEE international conference on information science & technology
12. 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, pp 3431–3440
13. Lowe DG, Lowe D (1999) Object recognition from local scale-invariant features. In: Proc. iccv
14. Loy CC, Gong S, Xiang T (2014) From semi-supervised to transfer counting of crowds. In: IEEE international conference on computer vision
15. Min L, Zhang Z, Huang K, Tan T (2009) Estimating the number of people in crowd- ed scenes by mid based foreground segmentation and head-shoulder detection. In: International conference on pattern recognition
16. Ojala T, Pietikinen M, Menp T (2002) Gray scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
17. Rodriguez M, Laptev I, Sivic J, Audibert JY (2011) Density-aware person detection and tracking in crowds. In: International conference on computer vision
18. Sam DB, Surya S, Babu RV (2017) Switching convolutional neural network for crowd counting
19. Shen Z, Xu Y, Ni B, Wang M, Hu J, Yang X (2018) Crowd counting via adver- sarial cross-scale consistency pursuit. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5245–5254
20. Sindagi VA, Patel VM (2017) Generating high-quality crowd density maps using contextual pyramid cnns. In: Proceedings of the IEEE international conference on computer vision, pp 1861–1870

21. Wu B, Nevatia R (2005) Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Tenth IEEE international conference on computer vision
22. Zhang Y, Zhou D, Chen S, Gao S, Yi M (2016) Single-image crowd counting via multi-column convolutional neural network. In: Computer vision & pattern recognition

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chuanrui Hu received his bachelor's degree in Automation from Anhui University of Technology, Maanshan, China, in 2016. He is pursuing her master degree of pattern recognition Anhui University, Hefei, China. He is currently developing techniques to solve large-scale semi-supervised segmentation problems. His research interests include semi-supervised learning, support vector machine, and algorithm.



Kai Cheng received his bachelor's degree in Automation from Wuhan Huaxia University of Technology, Wuhan, China, in 2017. He is pursuing her master degree of control engineering Anhui University, Hefei, China. He is currently developing techniques to solve saliency object detection problems. His research interests include semi-supervised learning, support vector machine, and algorithm.

Yixiang Xie received his bachelor's degree in Automation from Anhui University, Heifei, China, in 2016. He is pursuing her master degree of pattern recognition Anhui University, Hefei, China. He is currently developing techniques to solve large-scale person re-identification problems. His research interests include deep learning, support vector machine, and algorithm.

Teng Li received B.S. from University of Science and Technology of China (USTC) in 2001, M.S. from Institute of Automation, Chinese Academy of Sciences (CASIA) in 2004, and Ph.D. from Korea Advanced Institute of Science and Technolgy (KAIST) in 2010. He is currently a professor with Anhui University.