Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6664281, 14 pages
https://doi.org/10.1155/2021/6664281

Research Article

A Crowd Counting Framework Combining with Crowd Location

Jin Zhang,[1] **Sheng Chen**,[1] **Sen Tian**,[2] **Wenan Gong**,[3] **Guoshan Cai**,[4] **and Ying Wang**[5]

[1]*College of Informatica Science and Engineering, Hunan Normal University, Changsha 410081, China*
[2]*College of Mathematics and Statistics, Hunan Normal University, Changsha 410081, China*
[3]*Changsha Transportation Information Center, Changsha 410016, China*
[4]*Changsha Tianxia Yida Information Technology Co., Ltd., Changsha 410221, China*
[5]*School of Humanities and Management, Hunan University of Chinese Medicine, Changsha 410208, China*

Correspondence should be addressed to Jin Zhang; jinzhang@hunnu.edu.cn

Received 21 December 2020; Revised 29 December 2020; Accepted 4 February 2021; Published 17 February 2021

Academic Editor: Chi-Hua Chen

In the past ten years, crowd detection and counting have been applied in many fields such as station crowd statistics, urban safety prevention, and people flow statistics. However, obtaining accurate positions and improving the performance of crowd counting in dense scenes still face challenges, and it is worthwhile devoting much effort to this. In this paper, a new framework is proposed to resolve the problem. The proposed framework includes two parts. The first part is a fully convolutional neural network (CNN) consisting of backend and upsampling. In the first part, backend uses the residual network (ResNet) to encode the features of the input picture, and upsampling uses the deconvolution layer to decode the feature information. The first part processes the input image, and the processed image is input to the second part. The second part is a peak confidence map (PCM), which is proposed based on an improvement over the density map (DM). Compared with DM, PCM can not only solve the problem of crowd counting but also accurately predict the location of the person. The experimental results on several datasets (Beijing-BRT, Mall, Shanghai Tech, and UCF_CC_50 datasets) show that the proposed framework can achieve higher crowd counting performance in dense scenarios and can accurately predict the location of crowds.

1. Introduction

The crowd counting methods are used in videos and pictures to predict the number of people. For example, it's beneficial, especially in case of an emergency, such as Corona Virus Disease 2019. Otherwise, it can also be used to perform similar tasks, such as vehicle counting and cell counting under a microscope. Like other computer vision tasks, crowd counting also faces enormous challenges in terms of occlusion, background interference, and image distortion.

Many excellent models and algorithms are proposed to solve these problems in crowd counting. The methods for solving crowd counting can be classified into two categories: traditional methods and methods based on convolutional neural network (CNN). The conventional methods focus on carefully designed features extraction algorithms to solve this problem. However, the conventional methods are difficult to handle dense scenes. Due to the good performance of deep learning in various fields in recent years, the problem of crowd counting is increasingly being solved by CNN. CNN-based methods are easy to use and have better performance.

Crowd counting methods based on CNN consist of two categories: DM-based methods and detection-based methods. The DM-based method [1] first uses a normalized Gaussian kernel to represent the number of people, then predicts the DM through the CNN, and finally sums the DM to obtain the number of people. The detection-based method is to detect the number and location of the crowd by training a crowd detector. Compared with the detection-based methods, the DM-based methods have more robust to highly occluded scenes [2]. However, the DM-based methods lead to the following problems [3]: (1) higher the proportion of false positives and (2) loss of crowd location information.

As the crowd density increases, it is particularly important to study methods for dense scenes. However, most of

the current research methods only focus on the design of the network structure and ignore the fundamental problem brought by DM: "location information loss." Location information and the number of people are complementary to each other. Therefore, a new crowd detection and counting framework is proposed to solve this problem.

Our main contributions are as follows.

We propose a new network structure called ResNet-DC. It uses the ResNet [4], which performs well on classification problems, as backbone. It uses the deconvolution layer as upsampling. It is compatible with other powerful network structures so that we can migrate other network structures, and the structure is applied to both DM and PCM.

We propose a new PCM that links the crowd counting problem with the crowd detection problem. In dense scenes, PCM shows better performance than DM in the same network.

2. Related Work

For crowd counting, many powerful methods and algorithms are proposed. This section briefly describes two different methods: traditional methods and CNN-based methods.

2.1. Traditional Methods. In traditional crowd detection and counting methods, Chan and Vasconcelos [5] and Ryan et al. [6] proposed a regression-based method that predicts the number of people by first separating the background and then extracting features from the foreground. Lin and Davis [7] and Wang and Wang [8] proposed a detection-based method, which uses two consecutive video frame sequences. Idrees et al. [9] proposed an approach based on a carefully designed set of features: HOG. With HOG, head detection, Fourier analysis, and points of interest are integrated to avoid the disadvantages of a single feature. In traditional research methods, most research work focuses on carefully designed features to solve this problem. However, these methods are challenging to handle dense scenes or the image severely disturbed by the background.

2.2. Methods Based on CNN. With the development and application of deep learning, more and more research work is currently using CNN to solve crowd counting problems. At present, deep learning has been applied in many fields, such as traffic sign recognition [10], vehicle speed estimation [11], object tracking [12], and bus arrival prediction [13]. Compared with carefully designed solutions for feature extraction, CNN based methods are easy to use and have outstanding performance. CNN-based methods consist of two categories: the DM-based methods and the detection-based methods.

In DM-based methods, Zhang et al. [14] proposed a strategy based on DM in a cross-scene scenario, which randomly crops the image, divides the obtained features into two subtasks, and gets DM and the number of people through full connection. Ding et al. [15] proposed the use of a deeply recursive network (DR-ResNet). Unlike the previous ResNet, the ResNet block in DR-ResNet is constructed in different convolution, batch normalization (BN) [16], and rectified linear unit (ReLU) [17] order and then add to the input to adapt to the scene changes. When processing video data, the CNN-based method will only consider each video frame separately and ignore the temporal correlation of adjacent frames. Xiong et al. [18] highlighted a new variant of CNN, called CNN LSTM, which captures space and time dependencies. To obtain high resolution DM, Liu et al. [19] proposed a method to optimize the multicolumn convolution neural network by learning global features and recover the lost details in downsampling by deconvolution. To adapt to the characteristics of multiscale crowds, Zhang et al. [1] first proposed a method to solve the scale problem through different convolution kernel sizes. Sam et al. [20] proposed the use of a switching convolutional neural network, which maps image patches to specific CNN columns. Sang et al. [21] optimized the geometric adaptive Gaussian kernel function of SaCNN to generate a higher quality real DM. Kong et al. [22] proposed an adaptive attention mechanism method to automatically adjust the network structure through the crowd size.

In the detection-based methods, [2, 23, 24] all use Faster R-CNN [25] as the crowd detector. To overcome the limitations of pedestrian detectors, Saqib et al. [23] proposed a motion-guided filter (MGF), which uses temporal and spatial information among successive frames of video to recover lost details. The performance of the detector in dense scenes is improved, but this scheme is only applicable to video stream data. In dense scenes, due to the severe occlusion, Vora [2] and Kong et al. [22] detected the crowd heads, which increased the accuracy of detection. Vora [2] proposed faster R-CNN directly for binary classification tasks, to determine whether the detection frame is a human head and to reduce the number of anchor boxes according to the human head scale, speeding up the detection process. Basalamah et al. [24] and others proposed a Faster R-CNN-based scale driven convolutional neural network (SD-CNN) model to detect crowd heads and to solve the problem of different head sizes in video streams based on a scale map.

3. A New Framework for Crowd Detection and Counting Combining RESNET-DC and PCM

The framework includes two parts. (1) The first part is a full CNN, namely, ResNet-DC, which consists of backend and upsampling. (2) The second part is PCM, which contains information about the location. In this section, the proposed framework is introduced firstly. Then, two critical parts of the framework are described in detail. Finally, some training details are shown.

3.1. Framework Structure. As shown in Figure 1, there are three steps in the structure of the proposed framework for crowd detection and counting. The first step aims to extract input image features based on a CNN consisting of backend and upsampling. Backend shown in Figure 2 uses the ResNet to extract the features, and upsampling shown in Figure 3

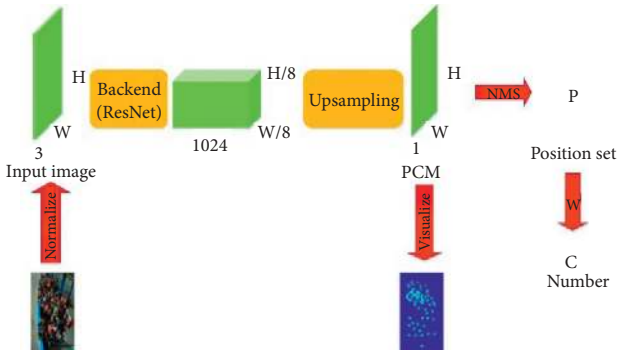


Figure 1: The structure of the proposed framework for crowd detection and counting.

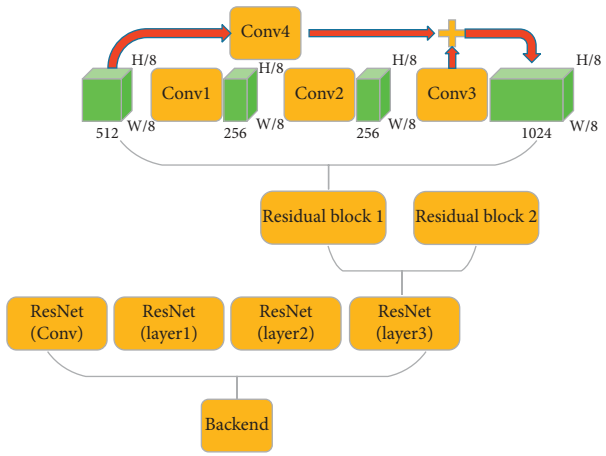


Figure 2: The network structure of backend.

uses the deconvolution layers to restore the feature map scale. The second step aims to predict high-quality PCM. The last step is to analyze the estimated position set P to get the number of people and location. To obtain the location information of the crowd, it is only necessary to perform nonmaximum suppression on PCM to get the location set. Therefore, we only need to count the location of the crowd to get the number of people.

3.2. ResNet-DC. The first part of the proposed framework is named as ResNet-DC. In ResNet-DC, backend extracts the features of the input image and reduces the input size by eight times, and upsampling restores the size of the feature map to obtain a high-quality PCM.

3.2.1. Backend. In this work, ResNet-18 [4] is used as the backbone network, which has outstanding performance in classification problems. In the backbone network, the deeper the network, the more increased the memory, training, and inference time. Due to the real-time nature of crowd detection, it is reasonable to use the first to third layers of ResNet. As the step size increases, the downsampling of the feature map increases. The step size of the residual block 1 in the third layer of ResNet is changed from two to one according to the crowd counting framework [26] to avoid severe loss of location information due to downsampling. Figure 2 shows the modified structure of the first residual block in layer three of ResNet. The detailed configuration is shown in Table 1. The subsequent residual blocks still retain the original design of ResNet. Under this setting, backend extracts the feature information of the original image and performs downsampling to obtain a feature map that is eight times smaller than the original.

3.2.2. Upsampling. In crowded scenes, excessive downsampling causes loss of feature information (especially location information). It is a feasible method to use the deconvolution layer to recover the feature information and obtain high-quality PCM. Deconvolution can be regarded as the inverse process of convolution and pooling. Long et al. [27] show that the deconvolution layer can recover more feature information than using convolution and bilinear interpolation. In this paper, the structure of upsampling is shown in Figure 3. It consists of two and three deconvolution layers. The first convolutional layer is responsible for compressing the channels of the feature map. The three deconvolution layers in the middle are accountable for upsampling the feature map to the original image size. The last convolutional layer is responsible for mapping the feature map to PCM. Table 2 shows configuration information for upsampling.

$$G_{\sigma_i}x, y = \begin{cases} \alpha e^{-\left((x-x_i)^2+(y-y_i)^2/2\sigma^2\right)}, & x_i, y_i - \frac{ksize}{2} \le x, \\ & y \le x_i, y_i + \frac{ksize}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

Under the above structure, ResNet-DC can restore the feature map reduced by backbone to the same size as the input. In this way, the predicted feature map will not ignore some peaks due to overlapping peaks.

3.3. Peak Confidence Map. PCM, an improvement over DM, is designed and compares with DM in this section. Then, a nonmaximum suppression algorithm is introduced to obtain crowd information from PCM.

3.3.1. Density Map. The density map design is based on [1, 28]. For a head position (x_i, y_i) in an image, a normalized Gaussian kernel function $G_{\sigma_i}x, y$ is generated in the neighborhood of its $ksize \times ksize$. $G_{\sigma_i}x, y$ can be expressed as follows: where α is the normalization factor so that $\sum G_{\sigma_i}x, y = 1$. σ_i is the variance of the Gaussian kernel of the ith head. In traditional DM, it is designed as a constant. To convert the marked points into a density function, the

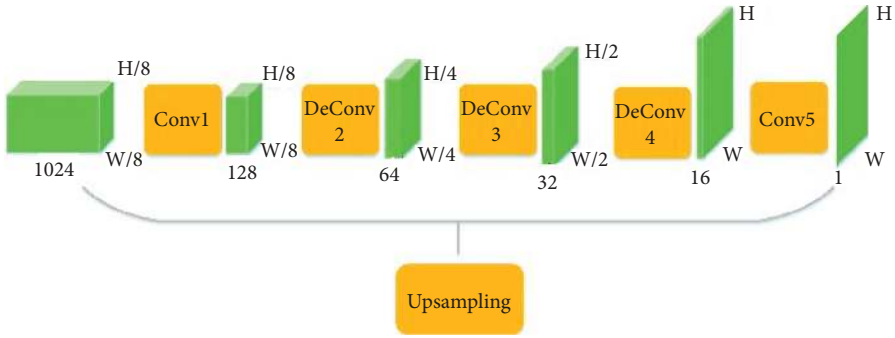


Figure 3: The structure of upsampling.

Table 1: The configuration of modified residual block 1 in ResNet18.

Layers name	Kernel	Stride	Input padding	Bias	BN	ReLU
Conv1	$256 * 1 * 1 * 512$	1	0	False	Yes	No
Conv2	$256 * 3 * 3 * 256$	1	1	False	Yes	No
Conv3	$256 * 1 * 1 * 256$	1	0	False	Yes	No
Conv4	$256 * 1 * 1 * 1024$	1	0	False	Yes	Yes

Table 2: The configuration of upsampling layer.

Layers name	Kernel	Stride	Input/output padding	Bias	BN	ReLU
Conv1	$128 * 3 * 3 * 1024$	1	1/-	Yes	Yes	Yes
DeConv2	$64 * 3 * 3 * 128$	2	1/1	Yes	Yes	Yes
DeConv3	$32 * 3 * 3 * 64$	2	1/1	Yes	Yes	Yes
DeConv4	$16 * 3 * 3 * 32$	2	1/1	Yes	Yes	Yes
Conv5	$1 * 1 * 1 * 16$	1	0/-	Yes	No	Yes

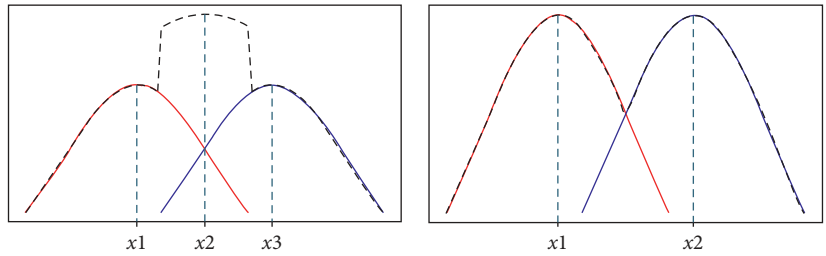
normalized Gaussian kernel function G_{σ_i} at different positions needs to be summed. The density function $F(x, y)$ can be expressed as follows:

$$M(x_i, y_i) = \max\left\{G_{\delta_i}(x, y), M(x_{i-1}, y_{i-1})\right\},$$
$$F(x, y) = M(x_N, y_N). \quad (2)$$

where $M(x_i, y_i)$ represents a density function that already contains i head positions and N represents the number of people in the ith image.

However, each head position is sampled in a 3D scene. Due to perspective distortion, different head sizes in the image are caused. Zhang et al. [1] found that the denser the crowd, the smaller the head size. To solve the problem of perspective distortion, [1] proposed a DM using an adaptive geometric Gaussian kernel based on the previous findings; that is, $\delta_i = \beta \overline{d}^i$. $\overline{d}^i$ represents the average of the distance between the ith head position and the k nearest heads, and $\beta = 0.3$ is obtained through experiments. Since $G_{\delta_i}(x, y)$ is normalized, each position corresponds to a Gaussian kernel function or adaptive geometric Gaussian kernel function with a sum of 1. By summing the pixels of the density function $F(x, y)$, the number of people can be obtained. However, due to the addition operation, false peaks may occur, which leads to the loss of position information. For example, there is a situation as shown in the left of Figure 4 (represented in one dimension), and the red and blue curves represent the Gaussian kernel function that transforms the position information of different people. It is easy to know that $x1$ and $x3$ represent different head positions, and the black curve can be obtained after the addition. Since a false peak $x2$ is generated at this time, it is impossible to determine which peak is the head position.

3.3.2. Peak Confidence Map. The different Gaussian kernel peaks correspond to the marked position of the head. In this paper, a design scheme for PCM that overcomes the shortcomings of location information loss is proposed. Unlike previous DM, the peak confidence function performs a maximum operation. PCM is defined in this paper as follows:



(a) (b)

Figure 4: The 1D DM and PCM.

$$G_{\sigma_i}x, y = \begin{cases} e^{-\left((x-x_i)^2+(y-y_i)^2/2\sigma^2\right)}, & x_i, y_i - \dfrac{ksize}{2}, \le x, \\ & y \le x_i, y_i + \dfrac{ksize}{2}, \\ 0, & \text{otherwise}, \end{cases}$$

$$M(x_i, y_i) = \max\left\{G_{\delta_i}(x, y), M(x_{i-1}, y_{i-1})\right\},$$

$$F(x, y) = M(x_N, y_N), \tag{3}$$

where $G_{\delta_i}(x, y)$ represents the Gaussian kernel corresponding to the ith head position, $M(x_i, y_i)$ represents a confidence function that already includes *i-head* positions, N represents the number of persons in the image, and σ_i is the ith heads correspond to the variance of the Gaussian kernel. Compared with DM, PCM no longer normalizes the Gaussian kernel because it uses the number of peaks to count the number of people and does not need to be summed like DM. The reason for named PCM is that (1) the peak represents the number and location of the crowd and (2) the closer to the head position, the higher the value. To some extent, it can reflect the confidence that a certain head position exists in PCM. Figure 4 shows the difference between PCM and DM. As shown to the right of Figure 4, if it is expressed in one dimension, the red and blue curves in the figure represent the Gaussian kernel functions corresponding to different heads positions. The black curve shows the results obtained by taking the maximum of different Gaussian kernel functions. As can be seen from the black curve, the two peaks exactly represent the head positions of different people. During the experiment, the peak confidence function was regressed to make the network produce different peaks at different people's head positions. By obtaining the extreme point from PCM to get the position of the peak, it is easy to know how many people will produce how many peaks.

According to the design method of PCM and DM, PCM and DM on Beijing-BRT [15], Mall [29], Shanghai Tech [1], and UCF_CC_50 [9] can be calculated. Figure 5 shows that there is not much difference between PCM and DM when the crowd is scattered. When the crowd is dense, the maximum value of PCM is at the head position of each person, and the location information and the crowd distribution can be calculated more precisely. But in DM, the denser the crowd, the greater the value, so the position information is lost.

In general, PCM and DM have the following differences. (1) DM takes the sum between Gaussian kernels, while PCM takes the maximum value between Gaussian kernels. (2) DM needs to normalize the Gaussian kernel, but PCM does not. (3) DM calculates the number of people by calculating the sum, and PCM calculates the position and the number of people by calculating the peak value.

3.3.3. Nonmaximum Suppression. Nonmaximum suppression aims at maximum local searching, that is, finding extreme points. In DM, due to the interference of false peaks, many incorrect positions will be detected by nonmaximum suppression method. So, it uses the regularized Gaussian kernel to calculate the number of people. This leads to the loss of location information. But in PCM, since each person's head corresponds to a peak, nonmaximum suppression becomes possible. The extreme point set P is calculated as follows:

$$P = \bigcup_{i=1}^{W} \bigcup_{j=1}^{H} \left\{\text{argmax}\left(F\left(x_i, y_j\right), \delta_4\right) > \vartheta\right\}, \tag{4}$$

where $F(x_i, y_j)$ denotes the (i, j)th pixel in PCM with the size of (W, H), δ_4 represents the four neighborhoods of pixels, ϑ is the confidence, and argmax denotes the subscript to get the maximum value. For each pixel of PCM, (7) compares it with its four domains. If the point is the maximum in four domains, then the pixel is the local maximum, that is, the extreme point. In other words, the head position P is a set: it is a local maximum and greater than the confidence.

3.4. Train Details. This section gives detailed training information on ResNet-DC. By using pretrained ResNet, ResNet-DC can quickly converge.

3.4.1. Label Normalize. The current work in [26] points out that a regression value will affect the performance of the network if a regression value is too small in DM. Considering the same effect on PCM, we multiply PCM by a factor of amplification. In this paper, we set the amplification factor

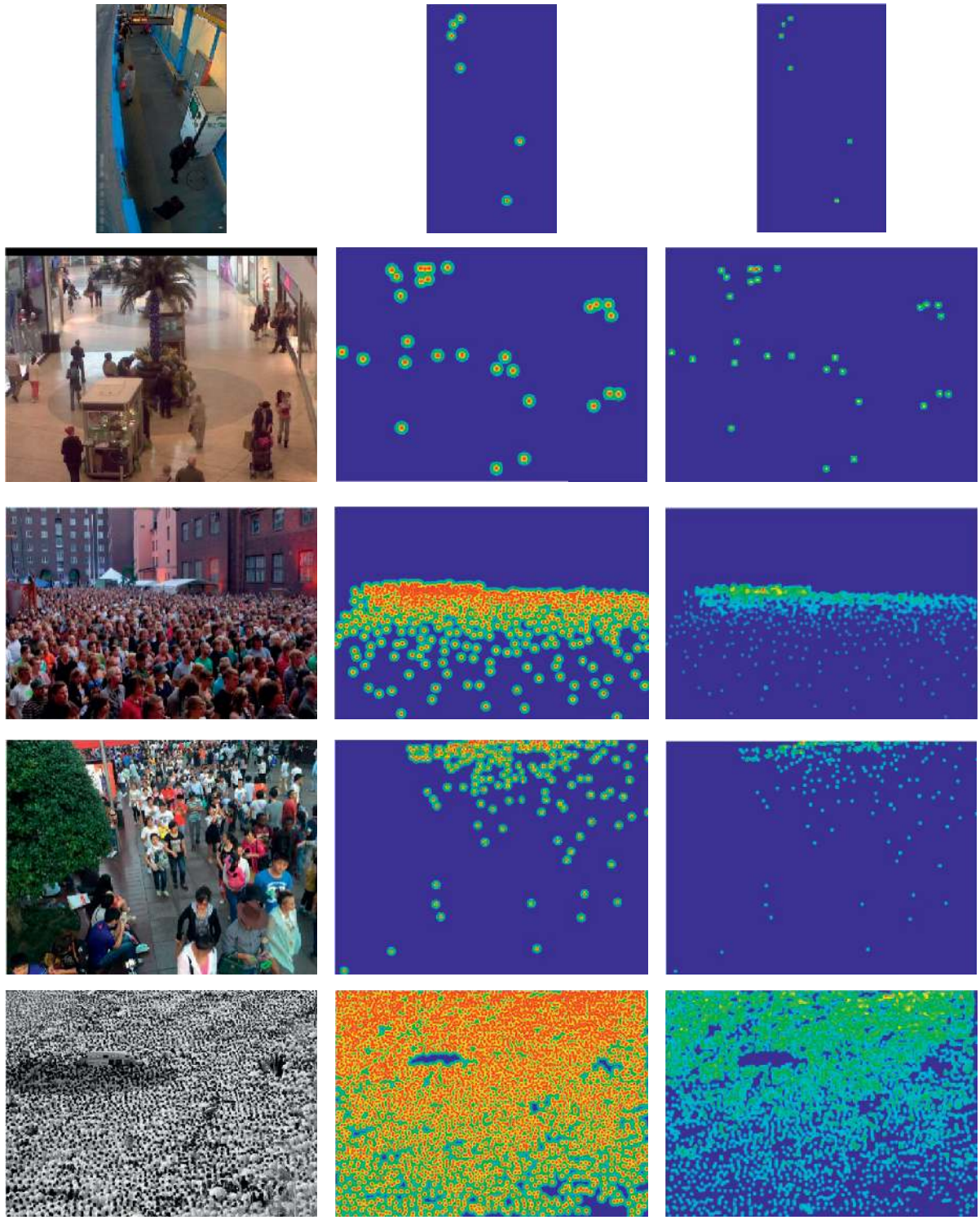


Figure 5: Comparison of density maps and peak confidence maps, sampled from Beijing-BRT (1st row), Mall (2nd row), Shanghai Tech Part A (3rd row), Shanghai Tech Part B (4th row), and UCF_CC_50 datasets (5th and 6th rows). The original pictures in the first column are sampled from different datasets. The picture in the second column represents the corresponding peak confidence maps. The picture in the third column represents the related density maps.

to 10. The reason for setting the magnification factor is that if the value of the PCM is too small, the network is easy to predict the wrong peak value, which is caused by the small difference between adjacent values. If the value of the PCM is too large, it is difficult for the network to converge, which is caused by the excessively large loss value.

3.4.2. Data Augment. The current work in [1] obtains nine times images by cropping at different positions. Since cropping may cause the loss of global information, in our experiments, we only flip the original image horizontally to obtain twice the image.

3.4.3. Loss Function. Most research work [1, 15, 20] uses the mean square loss to evaluate the error. In this paper, the mean square loss is also used. The MSE loss function is defined as follows:

$$L_{mse}(\theta) = \frac{1}{2N}\left|F(I_i;\ \theta) - G_i\right|^2, \tag{5}$$

where θ represents the parameters that ResNet-DC needs to learn, N represents the number of pictures, $F(I_i; \theta)$ represents PCM predicted by the ith input image I, and G_i represents the ground-truth PCM of the ith input image I. But when the mean square loss is only used, the network is biased towards more peaks predicted. Although the mean square loss can penalize the error between the ground-truth PCM and the estimated PCM, it ignores the relationship between adjacent pixels. Compared with DM, PCM has a stricter relationship with neighboring pixels. The reason for the extra peak is caused by ignoring the relationship between adjacent pixels.

In PCM, considering the importance of the relationship between adjacent pixels, a feasible solution is to calculate the difference between adjacent pixels. As we all know, the relationship between adjacent pixels can express important information. For example, the pixel values that are close to each other represent the same element, and the pixel values that are relatively different represent the boundaries of different elements. In order to express the above information, we use a convolution kernel with *kernel* = [[−1, −1, −1], [−1, 9, −1], [−1, -1, −1]]. The specific convolution kernel form is not important. We can use *kernel* = [[0, −1, 0], [−1, 5, −1], [0, −1, 0]] to achieve the same effect. Only the previous convolution kernel takes into account the values of the four corners. In this work, we use a convolution *kernel* = [[−1, −1, −1], [−1, 9, −1], [−1, −1, −1]] of size 3×3 to convolve with PCM to get the relationship between adjacent pixels. The loss is defined as follows:

$$L_{\text{ker}}(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left|F(I_i;\ \theta) * \text{kernel} - G_i * \text{kernel}\right|^2, \tag{6}$$

We use the kernel to convolve with PCM to obtain the difference value between the center point and its eight neighborhoods and then calculate the mean square error within the area. The total loss $L(\theta)$ can be calculated as follows:

$$L(\theta) = L_{mse}(\theta) + L_{\text{ker}}(\theta). \tag{7}$$

3.4.4. Learning Setting. According to transfer learning in [30] to accelerate model convergence, a straightforward way to train the ResNet-DC is used as an end-to-end structure. Backend is fine-tuned from a well-trained ResNet-18 [4]. For upsampling, the initial values come from a Gaussian initialization with 0.01 standard deviation. Using the Adam optimization algorithm, the learning rate is $5e-5$, and the weight decay rate is $1e-4$. The input image is regularized (mean and variance on the Imagenet dataset) and then trained on the dataset to predict PCM. At the same time, each iteration on the training set is verified on the validation set, and the best model in the validation set is retained.

4. Performance Evaluation

In this section, several datasets are used to evaluate performance. The crowd count evaluation metric and location evaluation metric are proposed. Based on the datasets and the metrics, the performance of different methods is compared and analyzed.

4.1. Dataset. Currently, the mainstream crowd count dataset includes Beijing-BRT [15], Mall [29], Shanghai Tech [1], and UCF_CC_50 [9]. In the framework, we performed experiments on the above four datasets, each of which is described as Table 3. In Beijing-BRT, we divided the training set and test set according to the criteria of [15]. In Mall, we divide the training set and test set according to the criteria of [18]. In Shanghai Tech, we divide the training set and test set according to the criteria of [14]. In UCF_CC_50, we use 5-fold cross-validation according to the standard of [9]. In these datasets, due to the different image resolutions of the Shanghai Part A and UCF_CC_50 datasets, we counted their average resolutions. We resized the image size so that it is closest to the average resolution and divisible by eight.

4.2. Evaluation Metric. According to the existing methods [1, 14], the mean square absolute error (MAE) and mean squared error (MSE) are used to evaluate the performance of crowd counting, which are defined as follows:

$$\begin{aligned}\text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|c_i - \widehat{c}_i\right|,\\ \text{MSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(c_i - \widehat{c}_i\right)^2},\end{aligned} \tag{8}$$

where N is the number of pictures, c_i is the number of people in the ith picture, and $\widehat{c}_i$ is the number of people predicted in the ith picture. To some extent, the mean square absolute error can be regarded as the accuracy of the prediction, and the mean square average error can be regarded as the generalization ability of the model. These two indicators are equally important. From the value of MAE and MSE, the lower the value, the higher the accuracy.

To quantitatively analyze the position performance, we use a method similar to object detection to evaluate the position performance as follows. (1) If a real position of the $S \times S$ neighborhood exists in the predicted position, we classify it as true positive. (2) If a predicted position does not belong to any of the real positions of the $S \times S$ neighborhood,

Table 3: Summary of the four datasets.

Datasets		Images	Count	Avg. density	Resolution	Avg. resolution	Resize
Beijing-BRT		1280	16,795	13.1	320 ∗ 640	—	—
Mall		2000	62,325	31.2	640 × 480	—	—
Shanghai tech	Part A	482	241,677	501.4	Different	868 ∗ 589	872 ∗ 592
	Part B	716	88,488	123.6	1024 × 768	—	—
UCF_CC_50		50	63,974	1279.5	Different	902 ∗ 653	904 ∗ 656

we classify it as false positive. Then, the standard Average Precision (AP) and Average Recall (AR) scores are calculated. In this experiment, S represents the allowed position error. We believe that due to the differences in manual marking, not all positions are accurately marked in the center of the human head, and there will be some errors. Therefore, when S is set to eight, it is reasonable to predict the position as the true positive.

4.3. Experimental Results and Analysis. In the experiment, we use the framework proposed in this paper to solve the crowd counting problem and crowd location prediction simultaneously. The experimental results on the above four datasets show that the proposed framework is not only suitable for dense scenes but also can predict the position of the crowd.

4.3.1. Counting Performance. In the experiment, we compared the crowd counting performance of DM and PCM. At the same time, we also compared it with other powerful algorithms. Tables 4–7 show the performance results of crowd counting on four different data sets. In DM, we use ResNet-DC to compare with other excellent algorithms, and the results show that the ResNet-DC has made slight progress in the Shanghai Tech part A (0.2 MAE) dataset and achieved good performance on other datasets. The results are acceptable because we used the simplest ResNet-18 as backend network. We can also use other deeper networks such as ResNet-32, ResNet-50, and ResNet-101. When we use PCM in ResNet-DC, we have performed excellent performance in Shanghai Tech Part A (2.33 MAE, 6.8 MSE) and good performance on other datasets.

4.3.2. Localization Performance. Because there are fewer experiments on localization on the crowd counting dataset, we only compare the AP and AR of different methods on the UCF_CC_50 dataset, as shown in Table 8. Compared with the current best algorithm SD-CNN [24], our approach is slightly worse on AP. But we have reached the best level in AR, and the improvement of 1.48 AP is better than the current best algorithm. We believe that this is because we only place the position with higher confidence (confidence is 0.5) as the location. Higher confidence leads to higher AP, but also lower AR. And because AP is soft, this leads to the degradation of MAE performance. Table 9 shows the position performance of our algorithm on the other three datasets. We found that although the performance of AP and AR gradually decreased with the increase of the crowd density, even on the worst-performing Shanghai Tech Part

Table 4: Counting performance of the different methods on Beijing-BRT.

Methods	Type	Position	Beijing-BRT	
			MAE	MSE
MCNN [1]	DM	No	2.24	3.35
FCNCC [31]	DM	No	1.74	2.43
ResNet-14 [15]	DM	No	1.48	2.22
DR-ResNet [15]	DM	No	1.39	**2.00**
ResNet-DC	DM	No	**1.36**	2.02
ResNet-DC	PCM	Yes	1.40	2.16

Table 5: Counting performance of the different methods on Mall.

Methods	Type	Position	Mall	
			MAE	MSE
CNNLSTM [18]	DM	No	2.24	8.5
ASA [22]	DM	No	2.3	3.0
MGF [23]	Detection	**Yes**	**1.89**	7.29
ResNet-DC	DM	No	2.33	**2.89**
ResNet-DC	PCM	Yes	2.49	3.14

Table 6: Counting performance of the different methods on Shanghai Tech.

Methods	Type	Position	Shanghai tech			
			Part A		Part B	
			MAE	MSE	MAE	MSE
MCNN [1]	DM	No	110.2	173.2	26.4	41.3
Switching CNN [20]	DM	No	90.4	135.0	20.0	33.4
MNCS [19]	DM	No	86.6	129.7	19.3	35.3
ASA [22]	DM	No	83.9	133.3	18.6	31.1
Sang et al. [21]	DM	No	75.84	124.9	11.0	**18.6**
ResNet-DC	DM	No	79.85	131.2	**10.8**	**18.6**
ResNet-DC	PCM	**Yes**	**73.51**	**118.1**	13.3	22.5

Table 7: Counting performance of the different methods on UCF_CC_50.

Methods	Type	Position	UCF_CC_50	
			MAE	MSE
Faster R-CNN [25]	Detection	**Yes**	592.09	672.19
MCNN [1]	DM	No	377.6	509.1
Switching CNN [20]	DM	No	318.1	439.2
MNCS [19]	DM	No	306.7	396.3
DA-Net [32]	DM	No	290.8	326.5
SD-CNN [24]	Detection	**Yes**	**235.74**	345.6
ResNet-DC	DM	No	286.3	415.0
ResNet-DC	PCM	**Yes**	254.78	**326.16**

Table 8: Localization performance of the different methods on UCF_CC_50.

Method	Type	UCF-CC-50	
		AP%	AR%
Faster R-CNN [25]	Detection	14.52	12.69
Kang et al. [3]	Detection	24.13	30.27
SD-CNN [24]	Detection	**45.67**	40.12
ResNet-DC	PCM	43.8	**41.6**

Table 9: Localization performance of ResNet-DC and PCM on Beijing-BRT, Mall, and Shanghai Tech.

Datasets		ResNet-DC + PCM	
		AP%	AR%
Beijing-BRT		65	66
Mall		66	63
Shanghai tech	Part A	59	59
	Part B	61	63

A, both AP and AR reached 59%. The result of four datasets shows that our algorithm can detect reliable locations even in dense scenes.

4.3.3. Result Analysis. Why is the performance of DM slightly better than PCM in sparse scenarios? Under the same network structure, the results show that in the sparse crowd scene (Beijing-BRT, Mall, and Shanghai Tech Part B), the design of DM is slightly better than PCM for crowd counting. We consider that this is since (5) has robustness for DM. In DM, it ignores the relationship between adjacent pixels. Even if there is a small amount of value prediction error, the impact on the crowd count is relatively small. PCM pays more attention to the comparison between adjacent pixels. Although (6) can mitigate the error value, it has not reached the optimal performance. Besides, we visualized the prediction results of PCM and DM under the same network structure. As shown in Figure 6, due to picture distortion, ResNet-DC loses information about people in the distance. Affected by the shooting environment, ResNet-DC lost the information of nearby people. For the missing information, PCM shows a lower confidence (lower than the confidence value 0.5), so PCM directly discards these values. Instead, the DM will add these values to the number of people. As a result, the predicted number of DM is closer to the true value than PCM.

Why is the performance of PCM better than DM in dense scenarios? Under the same network structure, the results show that in crowded scenes (Shanghai Tech Part A and UCF_CC_50), the crowd counting performance of PCM is significantly improved compared to the DM method. We also visualized the prediction results of PCM and DM under the same network structure. As shown in Figure 7, due to the defects of the convolutional network, the predicted picture in the dense scene is disturbed by the background (red rectangle). In PCM, since we define the peak value to be greater than the threshold value, nonmaximum suppression can filter out small activation values. In DM, these interference values are usually added to the number of people, resulting in the DM.

The method predicts a larger number of people. Besides, the results show that the network is generally interfered by occlusion in dense scenes, resulting in incorrect predictions (black rectangles) in dense areas. Because PCM combines position information, it is only sensitive to peaks. DM will directly add these false values to the number of people, which leads to the instability of the forecast results.

Why is PCM better than DM? First of all, due to design differences, PCM naturally contains location information, but DM does not. Secondly, since the peak value indicates the location of the crowd, PCM can ignore small activation values, thereby significantly reducing background interference. Conversely, DM adds these interference values to the crowd count. Finally, because PCM focuses on local maximums, it can ignore the second largest activation values generated in crowded places. Conversely, DM will also add the false activation values to the crowd count. We also visualized some of the results on the test set in Figure 8. Figure 8 shows that the predicted PCM generally has a high confidence level for the predicted crowd location on a dataset with low crowd density (Beijing-BRT, Mall, and Shanghai Tech Part B). On a dense crowd dataset (Shanghai Tech Part A and UCF_CC_50), the confidence level of the predicted PCM for the predicted crowd location is generally lower. As the crowd density increases, the peak confidence decreases. This phenomenon is consistent with people's intuitive feelings. At the same time, Figure 8 also shows that PCM can accurately predict the location of the crowd, which DM cannot do.

In general, PCM shows better performance than DM when faced with computer vision occlusion, background interference, and image distortion. Specifically, for occlusion and image distortion issues, PCM only considers the peak value. That is to say, even if there are overlapping or different-sized headers, PCM only needs to consider whether there is otherwise in the prediction result and does not need to consider the global information of the headers like DM. As for background interference, PCM can also filter out the interference information.

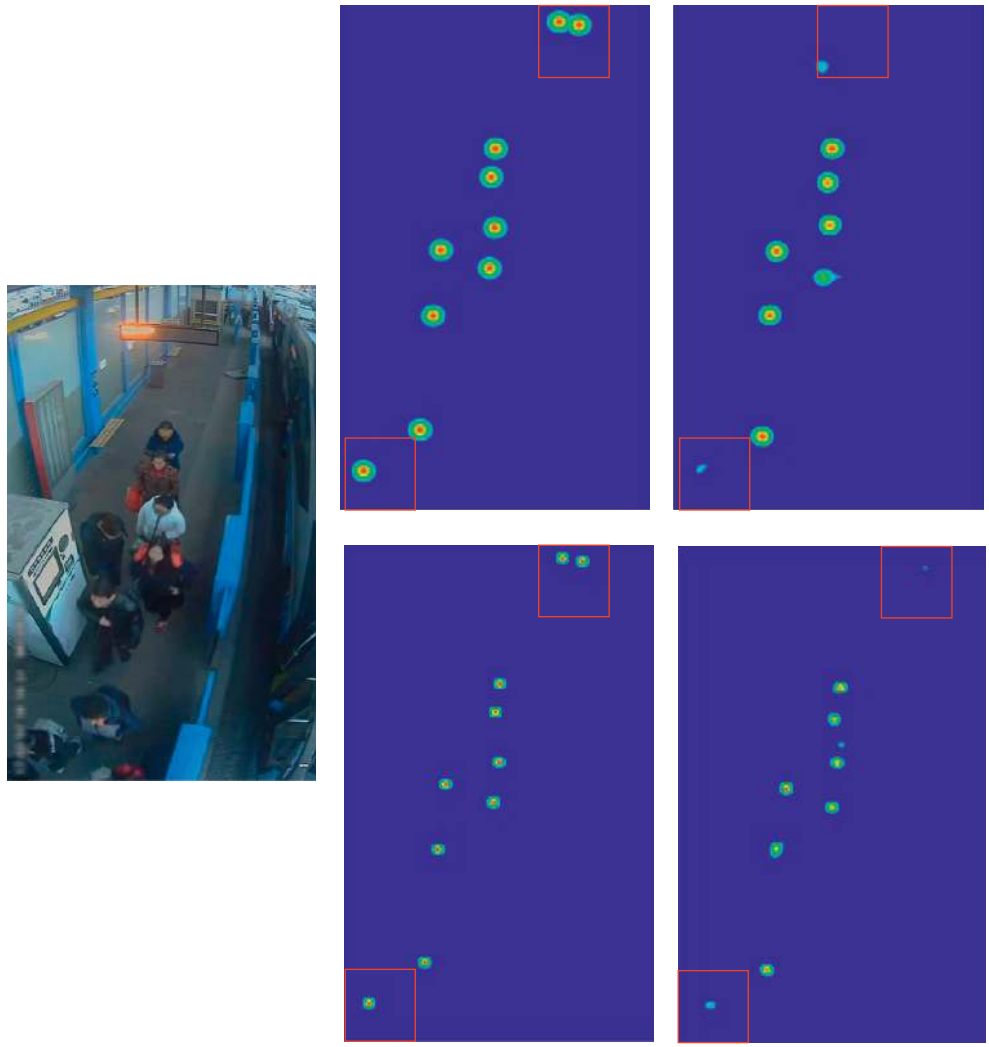


Figure 6: In the sparse scene, the comparison of predicted DM and PCM on ResNet-DC. The images sample from Beijing-BRT. The second column represents the true PCM or DM. The third column represents the predicted result. The first row represents PCM, and the second row represents DM.

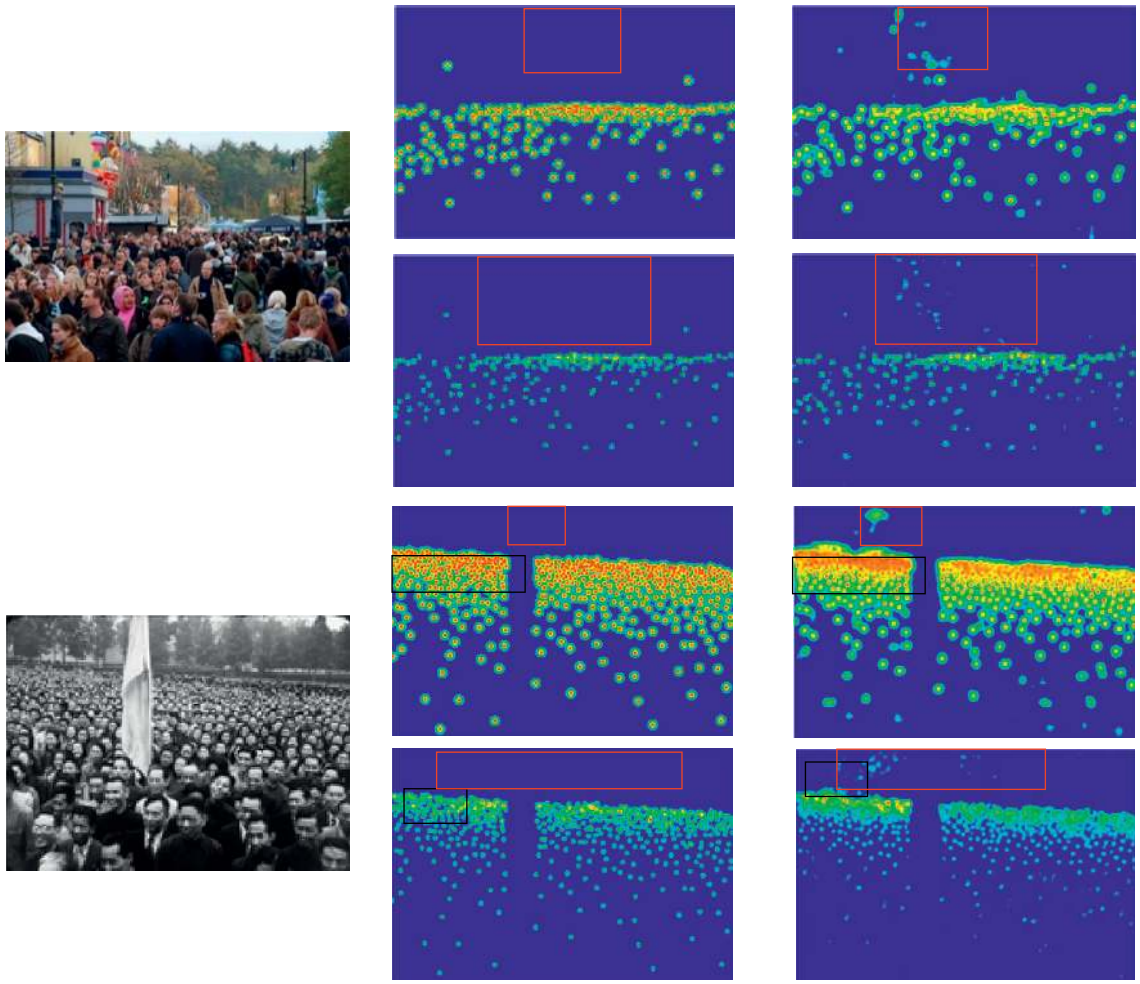


Figure 7: In dense scenarios, the comparison of predicted DM and PCM on ResNet-DC. The images sample from Shanghai Part A The 2nd column represents true PCM or DM. The 3rd column represents the predicted result. The 1st and 3rd rows represent PCM, and the 2nd and 4th rows represent DM.

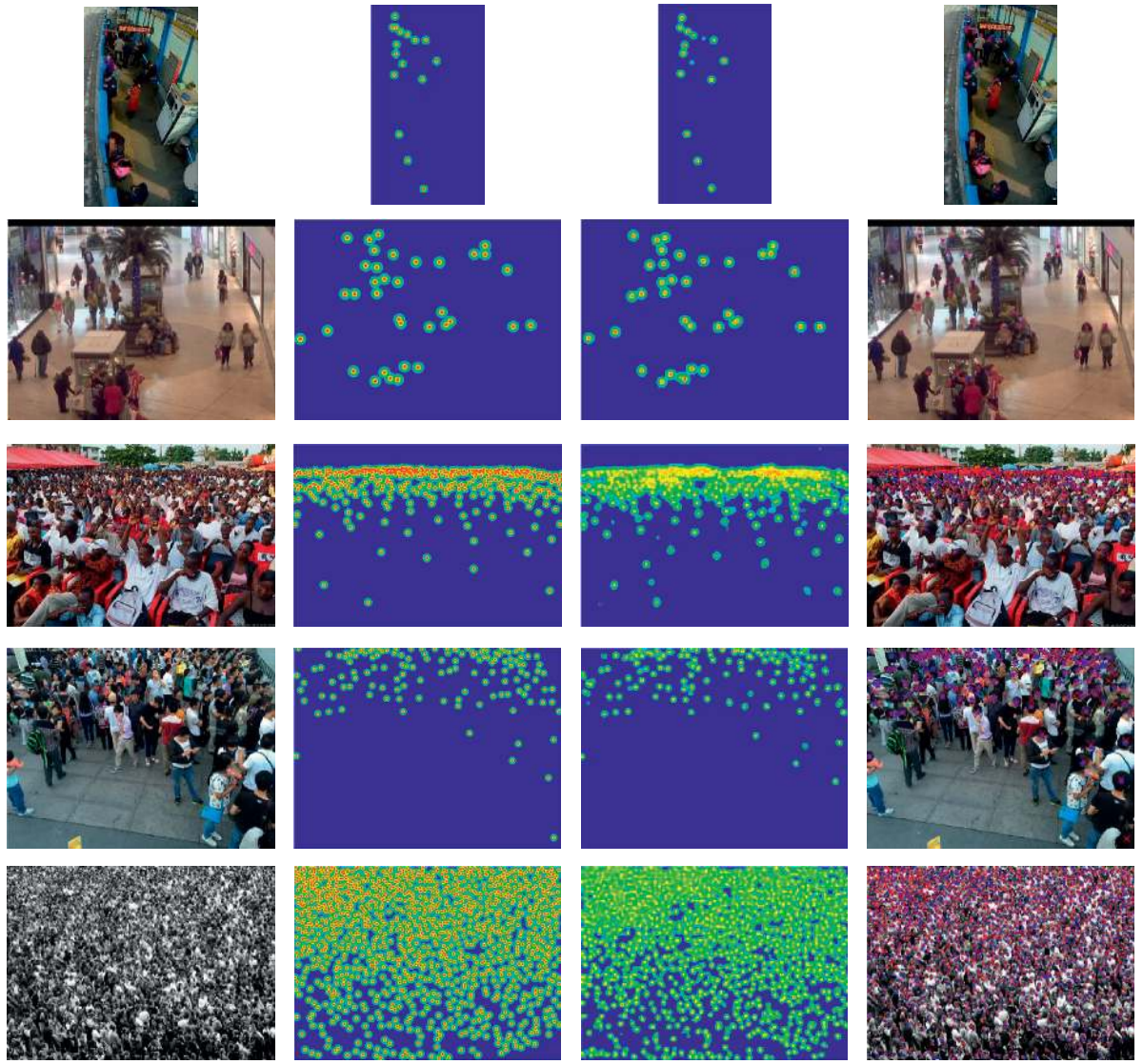


Figure 8: Results of sample images from Beijing-BRT (1st row), Mall (2nd row), Shanghai Part A (3rd row), Shanghai Part B (4th row), and UCF_CC_50 (5th row). The original pictures in the first column are sampled from different datasets. The pictures in the second column represent the corresponding ground-truth peak confidence map. The pictures in the third column represent the corresponding estimated peak confidence map. The pictures in the fourth column represents the ground-truth (indicating in red) and estimated (indicating in blue) position.

5. Conclusion

In this paper, a new framework is proposed to solve the problem of crowding detection and counting at the same time. The framework combines ResNet-DC with PCM to predict the number of people and the position of the person. ResNet-DC is a full CNN consisting of backend and upsampling. Backend is used as a feature extractor, and upsampling maps the extracted features into a high-quality PCM. The entire network is an end-to-end structure, and it is easy to migrate other excellent models to ResNet-DC. PCM retains the crowd distribution and location information. It can obtain position information through nonmaximum suppression and is also an effective method to solve background interference. Experimental results on four public datasets show that the proposed framework has good crowd counting performance and can even get accurate location information.

Data Availability

The related codes and data in the literature are released at https://github.com/Yuesheng321/RestNet-DC.git.

Conflicts of Interest

The authors declare that there are no conflicts of interest in the submission of this manuscript.

Authors' Contributions

The manuscript was approved by all authors for publication.

Acknowledgments

This work was supported by the education and research projects of Hunan Provincial Education Department (JG2018A012, XiangJiaoTong [2019] no. 291-410, no. 248-27, no. 370, [2020], no. 9, no. 90, and no. 233 HNKCSZ-2020-0122), the projects of the Ministry of Education of the People's Republic of China (201901051021), and the Science and Technology Progress and Innovation Project of Hunan Provincial Department of Transportation (no. 201927).

References

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas*, pp. 589–597, NV, USA, June 2016.

[2] A. Vora, "FCHD: a fast and accurate head detector," 2018, https://arxiv.org/abs/1809.08766.

[3] D. Kang, Z. Ma, and A. B. Chan, "Beyond counting: comparisons of density maps for crowd analysis task—counting, detection, and tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1408–1422, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[5] A. B. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 545–551, New York; NY, USA, September 2009.

[6] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression models," *Computer Vision and Image Understanding*, vol. 130, pp. 1–17, 2015.

[7] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 606–618, 2010.

[8] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3401–3408, Colorado Springs, CO, USA, June 2011.

[9] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2547–2554, Portland, OR, USA, June 2013.

[10] J. Zhang, Z. Xie, J. Sun, X. Zou, and J. Wang, "A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection," *IEEE Access*, vol. 8, pp. 29742–29754, 2020.

[11] C.-H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 265–267, 2020.

[12] J. M. Zhang, J. Sun, J. Wang, and X. G. Yue, "Visual object tracking based on residual network and cascaded correlation filters," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1–14, 2020.

[13] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[14] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–841, Boston, MA, USA, June 2015.

[15] X. Ding, Z. Lin, F. He, Y. Wang, and Y. Huang, "A deeply-recursive convolutional network for crowd counting," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1942–1946, Calgary, AB, Canada, April 2018.

[16] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," 2010, https://www.cs.toronto.edu/%7Efritz/absps/reluICML.pdf.

[18] F. Xiong, X. Shi, and DY. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *Proceedings of the International Computer Vision (ICCV)*, pp. 1861–1870, Venice, Italy, October 2017.

[19] Z. Liu, Y. Chen, B. Chen, L. Zhu, D. Wu, and G. Shen, "Crowd counting method based on convolutional neural network with global density feature," *IEEE Access*, vol. 7, pp. 88789–88798, 2019.

[20] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings fo the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4031–4039, Honolulu, Hl, USA, July 2017.

[21] J. Sang, W. Wu, H. Luo et al., "Improved crowd counting method based on scale-adaptive convolutional neural network," *IEEE Access*, vol. 7, pp. 24411–24419, 2019.

[22] W. Kong, H. Li, G. Xing, and F. Zhao, "An automatic scale-adaptive approach with attention mechanism-based crowd spatial information for crowd counting," *IEEE Access*, vol. 7, pp. 66215–66225, 2019.

[23] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein, "Crowd counting in low-resolution crowded scenes using region-based deep convolutional neural networks," *IEEE Access*, vol. 7, pp. 35317–35329, 2019.

[24] S. Basalamah, S. D. Khan, and H. Ullah, "Scale driven convolution neural network model for people counting and localization in crowd scenes," *IEEE ACCESS*, vol. 7, pp. 71576–71584, 2019.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection proposal networks," *Advances in Neural Information Processing Systems*, vol. 39, no. 6, pp. 91–99, 2015.

[26] J. Gao W, B. Zhao, D. Wang, C. Gao, and J. Wen, C3 framework: an open-source pytorch code for crowd counting," 2019, https://arxiv.org/abs/1907.02724.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[28] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proceedings of the Conference Advances in Neural Information Processing systems(NIPS)*, pp. 1324–1332, Vancouver, Canada, December 2010.

[29] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localized crowd counting," *BMVC*, vol. 1, no. 2, 3 pages, 2012.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[31] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," 2017, https://arxiv.org/abs/1612.00220.

[32] Z. Zou, X. Su, X. Qu, and P. Zhou, "DA-net: learning the fine-grained density distribution with deformation aggregation network," *IEEE Access*, vol. 6, pp. 60745–60756, 2018.