Hindawi
Scientific Programming
Volume 2021, Article ID 8831458, 10 pages
https://doi.org/10.1155/2021/8831458

Research Article

Crowd Counting Based on Multiresolution Density Map and Parallel Dilated Convolution

Jingfan Tang, Meijia Zhou, Pengfei Li, Min Zhang, and Ming Jiang

School of Computer Science, Hangzhou Dianzi University, Hangzhou 310000, China

Correspondence should be addressed to Meijia Zhou; zhoumiga@hdu.edu.cn

Received 19 August 2020; Accepted 11 January 2021; Published 20 January 2021

Academic Editor: Ferruccio Damiani

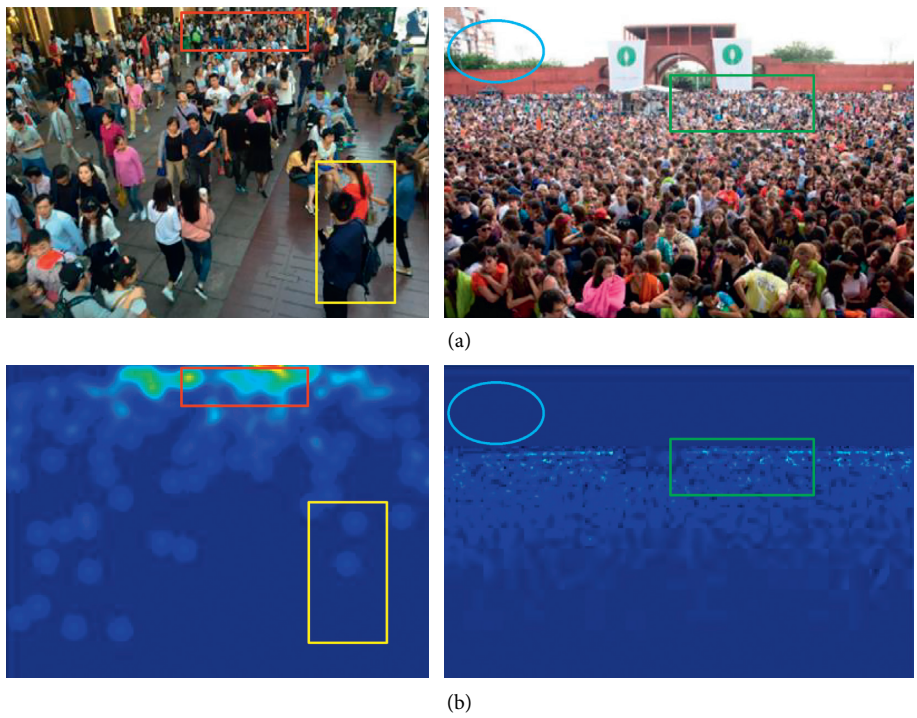
The current crowd counting tasks rely on a fully convolutional network to generate a density map that can achieve good performance. However, due to the crowd occlusion and perspective distortion in the image, the directly generated density map usually neglects the scale information and spatial contact information. To solve it, we proposed MDPDNet (Multiresolution Density maps and Parallel Dilated convolutions' Network) to reduce the influence of occlusion and distortion on crowd estimation. This network is composed of two modules: (1) the parallel dilated convolution module (PDM) that combines three dilated convolutions in parallel to obtain the deep features on the larger receptive field with fewer parameters while reducing the loss of multiscale information; (2) the multiresolution density map module (MDM) that contains three-branch networks for extracting spatial contact information on three different low-resolution density maps as the feature input of the final crowd density map. Experiments show that MDPDNet achieved excellent results on three mainstream datasets (ShanghaiTech, UCF_CC_50, and UCF-QNRF).

1. Introduction

As the phenomenon of crowd congestion is becoming serious, safety- and security-oriented tasks– such as public safety control and traffic safety monitoring– face huge challenges. Manual analysis of the degree of crowd aggregation not only cannot achieve high accuracy but also will perform low efficiently. In contrast, deep-learning-based methods are more applicable at present since their process not only eliminates manual efforts but also can analyze crowd aggregation accurately and quickly. Among them, crowd estimation at the pixel level through the crowd distribution density maps has achieved tremendous progress. A crowd density map is a kind of image label that can reflect the distribution of crowd heads by processing the head coordinate value through Gaussian convolution. As shown in Figure 1, the crowd in images mostly involves different distribution modes and aggregation features. The crowd distribution density map can obtain more accurate spatial information and more comprehensive image features in dense scenes, which brings about that the density estimation method can also be applied to vehicle control, bioecology research, and other cross-domain fields to share the advancement of this technology.

Early crowd counting methods rely on target detection or regression counting. The target-detection-based methods, such as Haar wavelet detection [1] and histogram direction gradient detection [2], are severely restricted by the occlusion of people and background clutter in the image, so they are only suitable for low-density scenes. The regression-based methods, such as counting through the features extracted by fusing Fourier analysis, head detection, and SIFT [3, 4] have broken the limitation of detection-based methods to some extent, but the expected results are often not achieved due to the difficulty of the regression process.

As the convolutional layer and pooling layer of Convolutional Neural Network (CNN) strengthen the relationship between pattern recognition and the context in the image, the density estimation methods of CNN are with strong learning ability. They have achieved high accuracy in dense scenes [5–7]. The accuracy of crowd counting mainly depends on the quality of the estimated density map which is



(a)

(b)

Figure 1: (a) Sparse crowd scene and its density map. (b) Dense crowd scene and its density map (the rectangular boxes represent different scales and the circle represents a complex background).

limited by the image scale. Since the convolution kernel of CNN owns a static size, heads of dynamic scales will worsen the network's performance, resulting in misjudgments and missing judgments. To solve this problem, the common methods are as follows: (1) introducing a multicolumn structure to estimate the crowd of different scales [8]; (2) introducing the idea of dilated convolution in the field from image segmentation [9]. This is a special convolution for extracting feature information of different scales, consisting of a 3*3 convolution kernel and a dilated parameter. By setting the dilated parameter to replace redundant branches of different sizes of convolution kernels, the computational cost of multiscale detection can be reduced; (3) applying different detection methods to regions of different scales in the image [10]. To generate a high-quality density map, spatial continuity should be ensured during the generation process so that the adjacent pixels in the output density graph can transition smoothly. Viresh et al. [11] proposed two-branch CNN architecture (ic-CNN) that generates a high-resolution density map from a low-resolution density map and an intermediate feature map. Since the low-resolution density map contains the spatial distribution information of the crowd, it can effectively improve the accuracy of prediction results.

In this paper, we design a novel crowd counting network named MDPDNet (Multiresolution Density maps and Parallel Dilated convolution Network). As shown in Figure 2, the parallel dilated convolution introduced by PDM can increase the receptive field and reduce the computational burden of the network under the condition of constant parameters, to speed up network training. MDM gradually generates three density maps of different resolutions and merges the spatial distribution information as an important feature input of the final density map. Meanwhile, to ensure that the resolution of the final density map is consistent with the original image, we use deconvolution to upsample the features to enrich the features' details. Inspired by ic-CNN, we use the minimization loss function to train the network's parameters, enhance the network's robustness, and optimize the network's iterative updating.

The main contributions of this model are as follows:

(1) We propose a novel crowd counting network (MDPDNet) to generate high-resolution density maps, count the crowd, and show the distribution accurately.

(2) Parallel dilated convolution is introduced to extract features on different receptive fields at high resolution to minimize the loss of multiscale information and fuse the crowd distribution information contained on the multiresolution density map to make the resulting high-resolution density map with close spatial correlation.

(3) MDPDNet shows better performance compared with several state-of-the-art approaches on three benchmark datasets.

The rest of this paper is organized as follows. In the next section, we review relevant work. In Section 3, we describe the proposed method, followed by its experimental evaluation in Section 4. The conclusion is presented in Section 5.

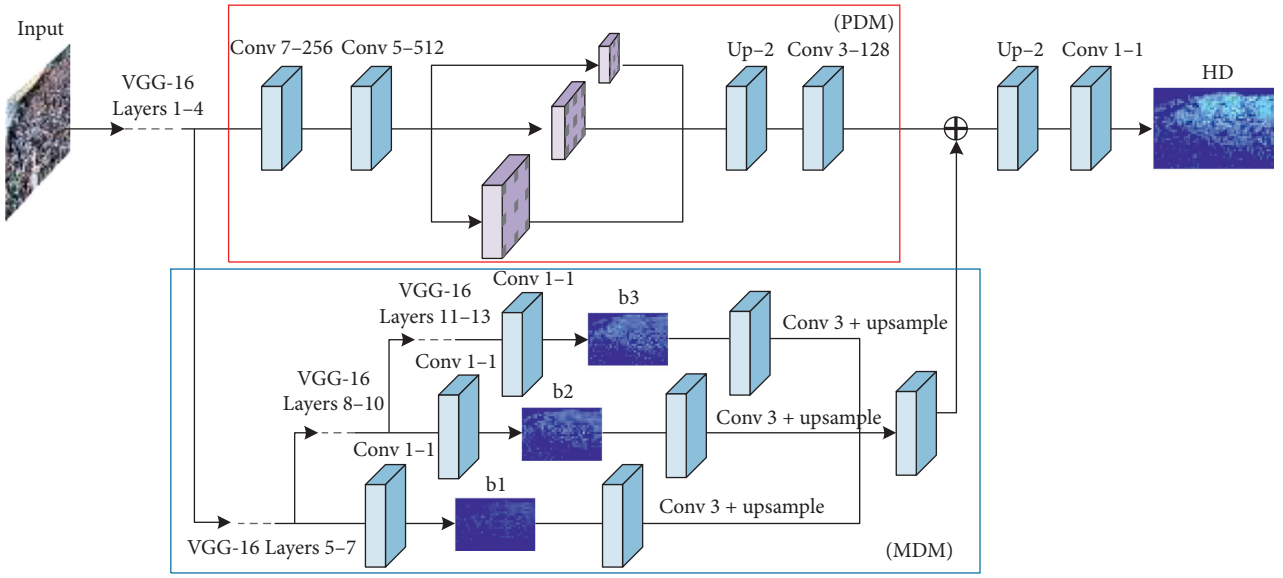


FIGURE 2: Crowd counting network model diagram based on multiscale density map fusion dilated convolution. (bx means multiresolution density map; HD means final high-resolution density map; MDM is responsible for retaining spatial contact information; PDM aims to expand receptive fields and further extract features; Up-2 represents the upsampling with a step size of 2; Conv7-256 represents the convolution kernel of $7 \times 7 \times 256$; Conv3 + Upsample means that there is a series of convolution and upsampling processes).

2. Related Work

The exploration of crowd counting methods has always been a hot research topic. Traditional crowd counting methods include target detection [1, 2, 12, 13] and feature regression [4, 14, 15]. Target-based detection usually applies a person or head detector to the image. For example, Dollar et al. [12] are the first to propose the sliding window detector to detect the crowd. Inspired by SIFT, Dalal Triggs [2] designed a directional gradient histogram HOG to calculate and count the gradient direction of the local region as a feature. Felzenszwalb et al. [13], by using feature pyramids to represent local feature mapping, overcame the shortcoming that global features are not adapted to occluded people to some extent. At present, although there is no ideal detection-based method in the face of high crowd concentration, the feature-regression-based method solves some occlusion problems by learning the relationship between local features. For example, Haroon et al. [14] extracted features based on points of interest through Fourier analysis and SIFT. Antoni and Vasconcelos [15] extracted features such as foreground and texture gradient to generate basic information. However, the method based on feature regression needs to establish the mapping relationship between crowd characteristics and crowd number first, which is a complicated process. Simultaneously, due to the serious background interference, it is easy to misestimate the crowd in the face of sparse scenes.

Given the powerful learning ability of CNN, it can not only optimize small-scale target detection by iterative learning but also improve detection accuracy while reducing detection computation. So it is commonly used to generate a crowd density map [16, 17]. Since Zhang et al. [8] proposed a simple multicolumn convolutional neural network structure (MCNN), each column of neural networks learns the features of the corresponding scale to adapt to images of different scales. The multicolumn convolutional structure is widely used in the crowd in the field of forecasting. The Switching-CNN proposed by Sam et al. [18] designed a selector to input image blocks to specific branches for feature extraction, further reducing the computational complexity of multicolumn networks.

Through experiments, Li et al. [19] found that MCNN can not only share features between internal branches but also create a redundant structure. To reduce the training complexity, they proposed CSRNet, which is a single-column network with a dilated convolutional layer at the back end. By setting the size of the dilation rate, CSRNet could provide different receptive fields to extract features of different scales of an image and successfully achieve the technology transformation from the image segmentation field to the crowd counting. On this basis, Ma et al. [20] proposed an atrous convolutions spatial pyramid network (ACSPNet), in which the convolutional blocks with different void rates integrate multiscale information and range through jump connection to improve scale perception. Although the application of dilated convolution greatly simplifies the network, it cannot reflect his superiority when there are objects with a lower resolution in the image. Zou et al. [21] designed a hierarchical scale recalibration network (HSRNet) for crowd gathering scenarios where scales vary dramatically. HSRNet can reconnect information across multiple scales by modeling contextual correlations. It is also very friendly when there are objects with lower resolution.

Besides, the method of estimating the crowd based on the density map fuses the spatial information of the image. However, the resolution of the image will decrease as deepening the network occurs, resulting in the fact that a large number of details of the spatial information will be lost, which limits the prediction accuracy of the final density map.

There are many upsampling methods currently used to solve the above problems. For example, ic-CNN [11] introduces a bilinear interpolation layer by using linear interpolation to fill pixel vacancies in two directions of the image to get a certain interpolation effect and operation speed. Zeiler and Fergus [22] proposed the up-pooling method, which only activates the value of the position coordinate where the maximum activation value is in the pooling process and sets the other values to 0, so the reconstructed image is discrete. SANet [23] used deconvolution to learn parameters during the upsampling process to improve the resolution of the features and make the density map contain finer spatial contact information.

Based on the study of the latest methods, we found that ic-CNN method based on a multiresolution density map can extract more abundant spatial information from crowd images when it is necessary to focus on solving the problem of spatial correlation of crowd targets and multiscale image. As for upsampling operations, compared with antipooling and bilinear interpolation, the deconvolution process can reflect the superiority of CNN. Therefore, we design a network model that generates multiresolution density maps in stages to strengthen the support of multiscale features, introduce void convolution to adapt to scale changes under high resolution, and simultaneously retain rich spatial connection information between features through deconvolution.

3. Proposed Approach

The distribution and scale of the crowd vary greatly between different images and even between different areas of the same image. To solve these two problems, we designed two modules for relevant optimization. As shown in Figure 2, one of them is a parallel dilated convolution module (PDM) for solving scale problems; the other is a multiresolution density map module (MDM) for solving spatial connection problems. In order to clearly describe the structure of this network, we also use simple symbols to represent the parameters at each stage in the prediction process. First, we mark the training image set as the set S; then, $S = \{(p_1, B_1, h_1) \ldots (p_n, B_n, h_n)\}$, where p_i represents the i-th test image, B_i is the set of multiresolution density map corresponding to p_i, h_i is the high-resolution density map predicted by p_i, and n represents the number of test images. This set contains all the to-be-predicted images involved in the training process and their corresponding density maps.

3.1. Overall Architecture of MDPDNet. We chose VGG-16 as the basic framework of this network. Its regular structure and strong adjustability can be flexibly applied to various models. Removing its final fully connected layer enables the network to adapt to the images' scale and provides a feature for MDM and PDM.

As shown in Figure 3, the first four layers of VGG-16 (Conv3-64, Conv3-64, Conv3-128, and Conv3-128) are introduced into the front end of MDPDNet as the basic feature extraction framework. Each layer of convolution is followed by the ReLU nonlinear activation function, which can effectively reduce the interdependence between parameters and enhance the nonlinear fitting ability of the network. After every two layers of convolution in VGG-16, a max-pooling layer with a step size of 2 is introduced to reduce the size of the picture and improve the utilization of network storage. After further features' extraction through parallel dilated convolution module (PDM) and multiresolution density map module (MDM), the outputs of the two modules are fused by connection operation and then pass through upsampling-2, Conv1-1, and output high-resolution density map with the original image $\{h, h_2 \ldots h_n\}$.

The mapping function of the final density map is

$$h_i = f_h(y_i, lf_i; \alpha_l, \alpha_h), \tag{1}$$

where $f_h(*)$ is the mapping function of the high-resolution module, h_i represents the high-resolution density prediction map corresponding to the picture p_i, α_l and lf are the weighted parameters and feature output of the MDM, and α_h and y_i are the parameter and feature output of PDM.

In the same way by which a high-resolution density map is generated based on low-resolution density maps, ic-CNN designs an iterative multistage extension. In each iteration, the network needs to fuse the predictions of all previous stages, which will increase the computation and complexity of the entire network. However, the multiple low-resolution density maps extracted from MDM by this network already contain rich contextual information of images, which is enough to provide a tight spatial correlation for the prediction of the high-resolution density map.

3.2. Multiresolution Density Map Module (MDM). The feature vectors obtained by convolution at different depths have different resolutions and have very different spatial relation information, so the resulting multiresolution density map contains tight spatial correlation. We derived this module at the front end of the backbone network. As shown in Figure 4, three groups of convolutional layers of VGG-16 are introduced layer by layer to form a branch network with different depths, and the feature vectors are simultaneously input into the branch network (L1, L2, and L3), further extract the context information of the image, and then simply output the corresponding low-resolution density map B_i ($B_i = \{b_{i1}, b_{i2}, b_{i3}\}$), which provides abundant spatial contact information for the important features lf_i of the high-resolution density map prediction task. Among them, b_{ij} is the low-resolution density map of the $j - th (1 \le j \le 3)$ depth corresponding to the test image p_i.

The mapping function of a low-resolution density map is as follows:

$$B_i = (b_{i1}, b_{i2}, b_{i3}) = (f_l(p_{i1}; \alpha_l), f_l(p_{i2}; \alpha_l), f_l(p_{i3}; \alpha_l)), \tag{2}$$

where α_l is the prediction parameter of the low-resolution density map branch; the feature input of the jth branch network is represented by p_{ij}; $f_l(*)$ represents the mapping function of the low-resolution density map.

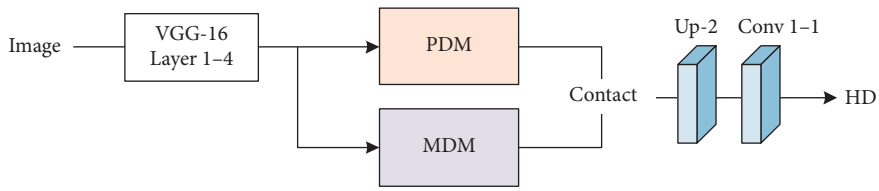


Figure 3: Overall framework of MDPDNet (PDM is the parallel dilated convolution module, MDM is the multiresolution density map module, HD is the final high-resolution density map, and Up-2 represents the upsampling with a step size of 2).

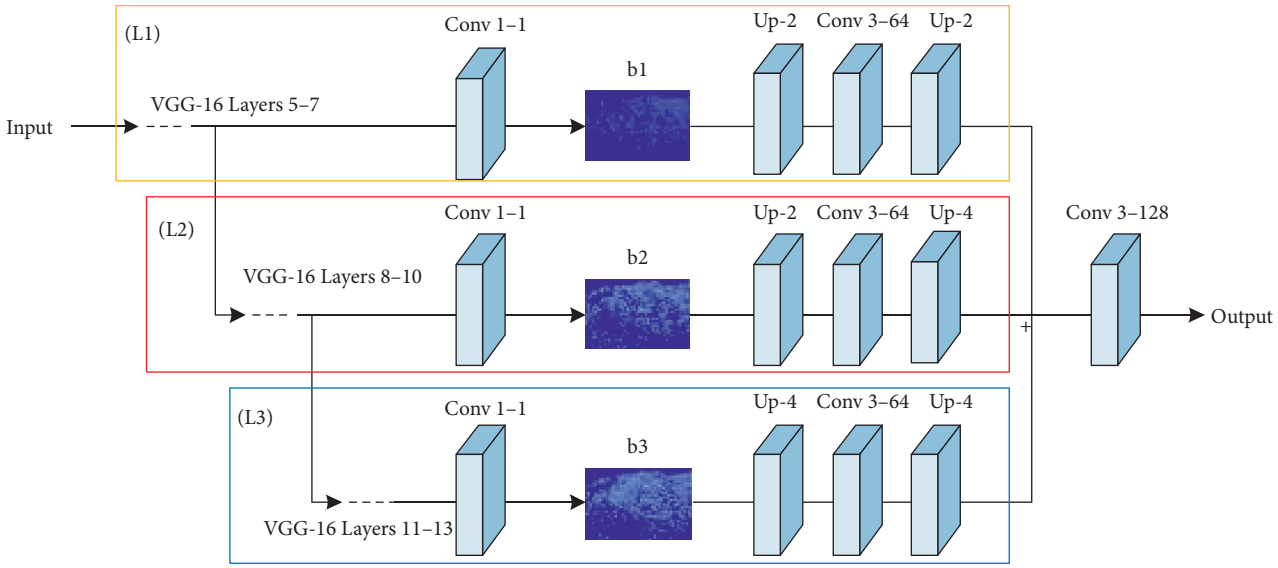


Figure 4: Structure configuration of MDM (L1, L2, and L3 represent three-branch networks with different resolutions); Conv1-1 denotes the convolution kernel of $1 \times 1 \times 1$, bx denotes the low-resolution density map, and Up-x denotes upsampling with a step size of x.

Since the output of the branch module will be used as an important feature input of the high-resolution density map, we use deconvolution to upsample the density map gradually in each branch to a high-resolution state to increase the sensitivity of the features to the information in the image. Finally, the important features lf_i are output through connection operation to enhance the hierarchical fusion of the multiresolution density map.

The convolution process of MDM is shown in Figure 4, which contains three columns of branch networks L1, L2, and L3. The detailed structure is shown in Table 1: L1: 5–7-layer convolution of VGG-16 (Conv3-256, Conv3-256, and Conv3-256), Conv1-1, upsampling-2, Conv3-64, upsampling-2, and output of the feature lf_1; L2: 8–10-layer convolution of VGG-16 (Conv3-512, Conv3-512, and-Conv3-512), Conv1-1, upsampling-2, Conv3-64, upsampling-4, and output of the feature lf_2; L3: 11–13-layer convolution of VGG-16 (Conv3-512, Conv3-512, and Conv3-512), Conv1-1, upsampling-4, Conv3-64, upsampling-4, and output of the feature lf_3. Finally, lf_1, lf_2, and lf_3 are connected by connection operation and convolved with conv3-128 to get the lf output of MDM back into the backbone network.

3.3. Parallel Dilated Convolution Module (PDM). This module is located at the back end of the backbone network. Compared with the redundant structure of a multicolumn network, the parallel dilated convolution constructed by this module only needs to configure convolution kernels with different dilation rates, which will increase the receptive field to further dig deeper into the image, thereby solving the problem that ordinary convolution cannot adapt to the scales, strengthening the robustness of the network.

Figure 5 shows the learning process of features in the PDM module. The most important part here is the three parallel dilated convolutions. The feature will be input into the parallel structure after Conv7-256 and Conv5-512 successively to extract the multiscale information of the image. In this module, we set the size of the parallel convolution kernels to 3×3, dilation rate to 1, 2, and 5, respectively, and the number of their channels to 64 (namely, Conv3-64-1-256, Conv3-64-2-256, and Conv3-64-5-256). Next, PDM will connect the extracted multiscale information through connection operation and use deconvolution for upsampling with a step size of 2 (Up-2) to increase its pixel level and after a further convolution (Conv3-128) to change the number of channels to obtain PDM's output.

Let y_i be the feature output of the input image p_i in PDM, and then the mapping process of parallel dilated convolution is

$$y_i = \text{Concat}\left(y_{i\_c}\right), \quad c = 1, 2, 5, \tag{3}$$

where c represents the dilation rates of the convolution kernel and y_{i_c} represents the feature output of a single convolution

Table 1: Detailed architecture of MDPDNet convolutional layer.

Layers	Backbone			
Input	Image			
1	VGG-16 layers 1–4			
Module	PDM	MDM		
2	Conv7 × 7 × 256	VGG-16 layers 5–7		
3	Conv5 × 5 × 512	Conv1 × 1 × 1 (b1)	VGG-16 layers 8–10	
4	Conv3-64-1-256	Upsampling-2	Conv1 × 1 × 1 (b2)	VGG-16 layers 11–13
5	Conv3-64-2-256		Conv3 × 3 × 64	Conv1 × 1 × 1 (b3)
6	Conv3-64-5-256	Conv3 × 3 × 64	Upsampling-4	Upsampling-4
7				Conv3 × 3 × 64
8	Upsampling-2			Upsampling-4
9	Conv3 × 3 × 128	Conv3 × 3 × 128		
10		Upsampling-2		
11		Conv1 × 1 × 1		
Output		HD		

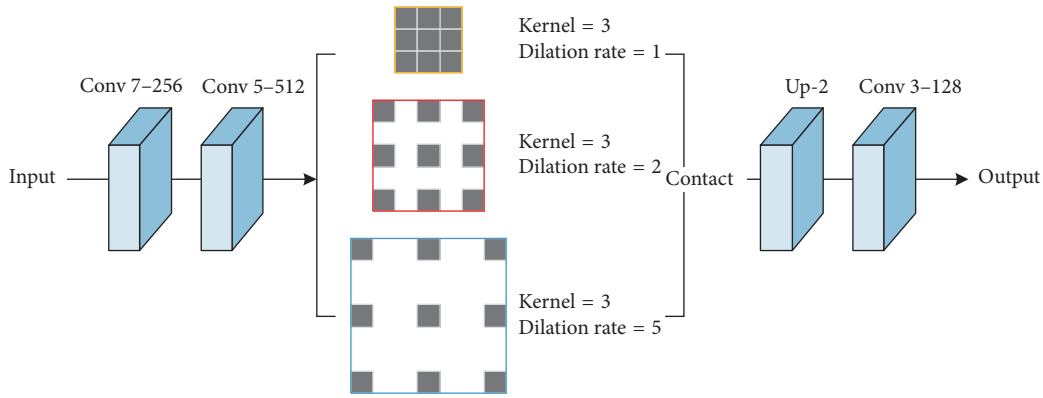


Figure 5: Structure configuration of PDM (kernel represents the size of the convolution kernel, dilation rate = 5 represents the size of the convolution's dilation rate which is 5, Conv7-256 represents the convolution kernel of 7 × 7 × 256, and Up-2 represents the upsampling with a step size of 2).

at different dilation rates, and its two-dimensional expression is

$$y_{i\_c} = \sum_{3} x[i + c^{*} 3] f[3], \tag{4}$$

where $x[i]$ represents the feature input of the image p_i for dilated convolution and $f[3]$ is a dilated convolution kernel of 3∗3. When $c = 1$, it is a special case of dilated convolution, namely, ordinary convolution kernel. Therefore, different sizes of receptive fields can be obtained by setting different dilation rates.

3.4. Training Details. We implemented our designed model using the PyTorch framework [24], and at the training stage, we define a fixed learning rate 10^{-5}, optimizer SGD, and training iteration number 2k. The parameters α_l and α_h are initialized to 10^{-2} and 10^{2}. Although MDPDNet has increased the receptive field as much as possible, the missed or wrong detections are inevitable due to the various image scales caused by the uneven distribution of crowds and inconsistent angles in crowded scenes. Therefore, we calculate the structural similarity between the predicted density map and the corresponding real density map by minimizing the loss function, thereby adjusting the parameters to enhance the robustness of the training stage against noise and optimize the iterative updating process. Our loss function is shown in

$$L(\alpha_l, \alpha_h) = \frac{1}{n} \sum_{i=1}^{n} L\left(f_h(P_i, B_i; \alpha_l, \alpha_h), D_{gt}\right), \tag{5}$$

where α_l and α_h are the mapping parameters of the two-branch modules, which are used to gradually optimize and adaptively adjust the proportion between the two-branch modules; D_{gt} represents the actual number of people marked manually.

4. Experiments

We conducted relevant training and testing on three public datasets [8, 14, 25] and obtained the expected result. In this section, we will introduce the evaluation indicators, analyze the branches of our model through the ablation study, and show the comparison of our proposed method with several recent state-of-the-art methods on these three datasets.

4.1. Evaluation Indicators. In the field of crowd counting, testing metrics that are produced by benchmarks include mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE). In this experiment, we chose MAE and MSE to evaluate the average-case performance on the test dataset since they can describe the stability of the model more comprehensively. The evaluation method is shown in

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N}|d_i - \overline{d}_i|, \\ \mathrm{MSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}|d_i - \overline{d}_i|^2}, \end{aligned} \tag{6}$$

where N represents the number of test images; $\overline{d}_i$ represents the actual number of test images p_i by artificial marks; d_i represents the number of people estimated by p_i through the network, and it is obtained by integrating the corresponding crowd density map.

4.2. Ground Truth Generation. For the density map to be able to adapt to various conditions of the crowd image, it can be expressed as $F(x)$ with N heads. The calculation method of $F(x)$ is to convolve the delta function $\delta(x - x_i)$ with a Gaussian Kernel $G_{\sigma_i}(x)$ normalized to 1:

$$F(x) = \sum_{i=1}^{N} \delta\left(x - x_i\right) * G_{\sigma_i}(x), \text{ with } \sigma_i = \beta \overline{d^i}, \tag{7}$$

where x_i represents the head per pedestrian on the pixel level; σ_i is the crowd distribution of all the images in the dataset; β is a constant; and $\overline{d^i}$ represents the average distance of k nearest neighbors of the target.

In our experiments, we follow the configuration in CSRNet [19]. Different datasets correspond to different σ of Gaussian kernels, and the settings are shown in Table 2. Certain parameters are set to fixed values ($\beta = 0.3$ and $k = 3$).

4.3. ShanghaiTech Dataset. ShanghaiTech dataset [19] is one of the general datasets in the field of crowd density statistics. It contains two parts, Part A and Part B. Herein, it covers several crowd gathering scenes of different scales and sizes. Part A is a collection of extremely crowded images and annotations, which consists of 300 training diagrams and 182 test diagrams. Part B collects a slightly sparse scene of the crowd, which consists of 316 training diagrams and 400 test diagrams and their annotations. Part A and B form a cross-scenario contrast.

The experiment result, shown in Table 3, describes that the MDPDNet can exert its accurate prediction ability in highly crowded scenes, which means that the algorithm we design performs as we expect. From the comparison experiment (just PDM or just MDM), it shows that when MDPDNet is predicted by only using the PDM module on the Part_A dataset (high crowd concentration), the MAE is reduced by 3.7%. Similarly, when MDPDNet is predicted by only using the MDM module on the Part_A dataset, the MAE is increased by 2.2%.

Table 2: The setups for different datasets.

Dataset	Parameter settings
ShanghaiTech Part_A [19]	$\sigma = 4$
ShanghaiTech Part_B [19]	$\sigma = 15$
UCF-QNRF [25]	Geometry-adaptive kernels
UCF_CC_50 [14]	Geometry-adaptive kernels

Table 3: Experimental results on ShanghaiTech dataset.

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
MCNN [8]	110.2	173.2	26.4	41.3
CSRNet [19]	68.2	115.0	10.6	16.0
Ic-CNN [11]	68.5	116.2	70.7	16.0
MBTTBF-SCFB [25]	60.2	**94.1**	8.0	15.5
HSRNet [21]	62.3	100.3	**7.2**	**11.8**
MDPDNet (this paper)	**59.8**	99.7	9.3	15.2
Just PDM (this paper)	63.5	105.2	10.1	15.9
Just MDM (this paper)	62.0	101.4	9.6	15.4

The opposite of the extremely crowded scene is that when faced with a slightly sparse crowd, because our network is essentially based on the idea of regression, there is a gap between the predicted results and the latest methods like HSRNet. However, because HSRNet heavily relies on detection and regression, the additional judgment steps have to be introduced to ensure the first-step image preprocessing, which increases complexity overhead.

4.4. UCF-QNRF Dataset. The UCF-QNRF dataset [25] contains rich crowd aggregation images, in which there are 1535 high-resolution images, 1201 images for training and 334 images for testing. No matter whether the image captures a sparse or dense crowd gathering scene, each one is annotated with a large amount of information. In this dataset, there are nearly 1.25 million annotations in total, which ensures strong reliability. According to the comparison results shown in Table 4, it can be seen that the crowd estimation accuracy produced by MDCNet improves 5.9% compared with other existing methods, such as MBTTBF-SCFB. Besides, when MDPDNet only uses the PDM module or the MDM module to perform individual prediction, the MAE improves about 3.0% and 9.8%, respectively.

4.5. UCF_CC_50 Dataset. The dataset [14] is composed of 50 instances: images and manual labeling information. Therein, the crowded images of people are with different perspectives, densities, and scenarios. Since the total number of instances in this dataset is small, it is a big challenge for training the network. The experiment results, shown in Table 5, illustrate that the MDPDNet we proposed greatly improves MAE and MSE metrics by 5.5% and 3.2%, compared with other classic models such as ic-CNN.

Table 4: Experimental results on UCF-QNRF dataset.

Method	MAE	MSE
MCNN [8]	277.0	426.0
Switch-CNN [18]	228.0	445.0
MBTTBF-SCFB [25]	**97.5**	165.2
MDPDNet (this paper)	100.4	**159.3**
Just PDM (this paper)	113.4	182.7
Just MDM (this paper)	120.2	201.3

Table 5: Experimental results on UCF_CC_50 dataset.

Method	MAE	MSE
MCNN [8]	377.6	509.1
Cascaded-MTL [25]	322.8	397.9
Switch-CNN [18]	318.1	439.2
ACSCP [26]	291.0	404.6
SaCNN [27]	314.9	424.8
CSRNet [19]	266.1	397.5
Ic-CNN [11]	260.9	365.5
MDPDNet (this paper)	**255.4**	**362.3**

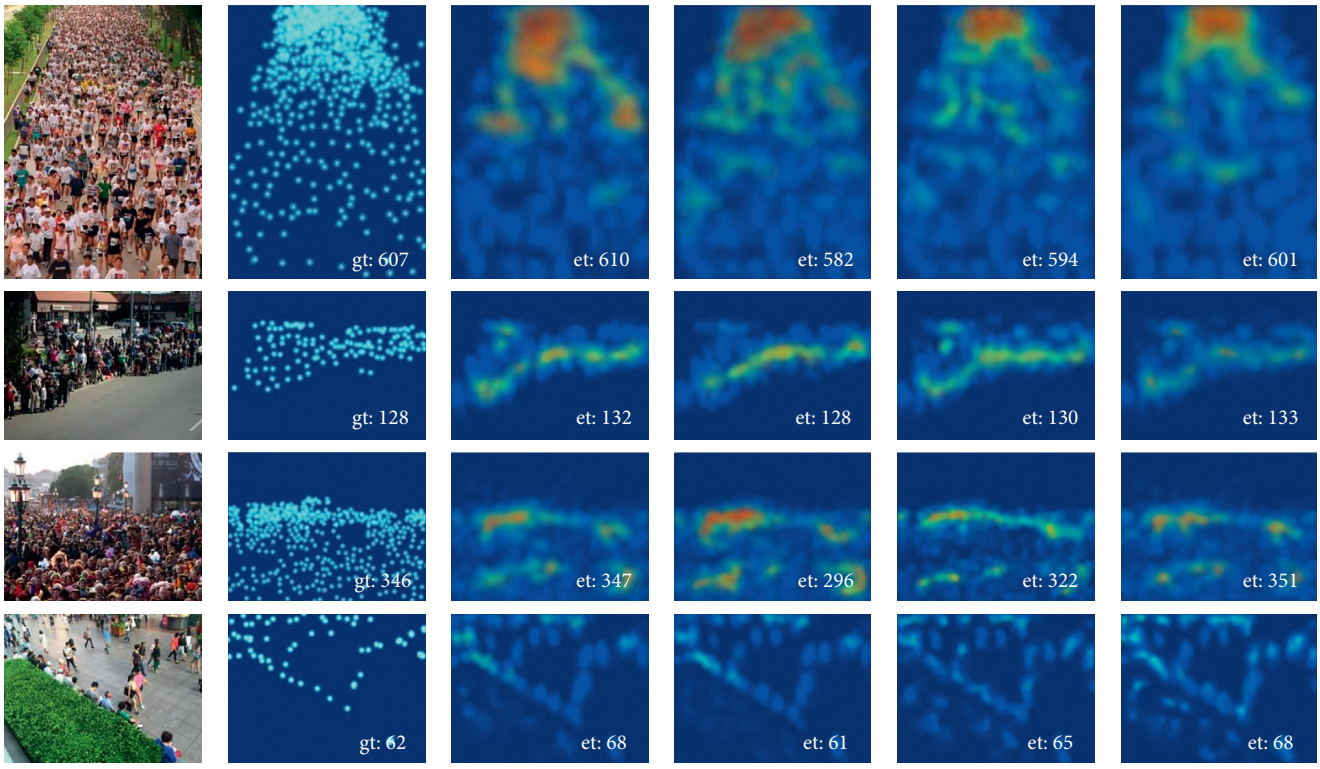


Figure 6: Experimental results. (a) The original image; (b) the manual annotation image; (c) the high-resolution density map produced by the MDPDNet; (d, e, and f) the three low-resolution density maps with L1, L2, and L3, respectively. We can see that the higher resolution the branch has (L3 > L2 > L1), the more accurate the positioning of the small-scale portrait of the image is.

4.6. Density Map. MDPDNet not only memorizes the final high-resolution density map but also records the density map of the low-resolution branch during the prediction process. As shown in Figure 6, there are four sets of crowd image inputs with different scales and different spatial distributions and their corresponding multiresolution density map outputs. From it, we can see that the final output of the MDPDNet model (column (c)) has reached a very high accuracy rate, which is very close to the manual labeling (column (b)). Particularly, in superdense crowd scenes (the third row), the accuracy and mapping quality of high-resolution density maps reach the expected level we expected.

5. Conclusions

In this paper, we propose a network structure named MDPDNet for crowd density estimation based on multiscale density map fusion dilated convolution. This structure can extract the spatial contact information of different features in one iteration by performing density analysis on the features at different scales. On this basis, the multiscale features that are processed by parallel dilated convolution are further fused to generate the final high-quality crowd density estimation map with a close correlation. The comparative experiments on several classic datasets (ShanghaiTech, UCF-QNRF, and UCF_CC_50) show that MDPDNet we proposed is of high accuracy and strong robustness.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding this paper.

Acknowledgments

This work was supported by Zhejiang Provincial Technical Plan Project (no. 2020C03105).

References

[1] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, San Diego, CA, USA, July 2005.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, Corfu, Greece, September 1999.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[8] Y. Zhang, D. Zhou, S. Chen et al., "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 589–597, Las Vegas, NV, USA, October 2016.

[9] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, https://arxiv.org/abs/1511.07122.

[10] B. Wei, Y. Yuan, and Q. Wang, "MSPNET: multi-supervised parallel network for crowd counting," in *Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2418–2422, Barcelona, Spain, May 2020.

[11] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 270–285, Munich, Germany, September 2018.

[12] P. Dollar, C. Wojek, B. Schiele et al., "Pedestrian detection: an evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2011.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester et al., "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[14] H. Idrees, I. Saleemi, C. Seibert et al., "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2547–2554, Portland, Oregon, June 2013.

[15] A. B. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, pp. 545–551, Kyoto, Japan, September 2009.

[16] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1324–1332, 2010.

[17] V. Q. Pham, T. Kozakaya, O. Yamaguchi et al., "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*, pp. 3253–3261, Las Condes, Chile, December 2015.

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

[19] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1091–1100, Salt Lake City, UT, United States, June 2018.

[20] J. Ma, Y. Dai, and Y.-P. Tan, "Atrous convolutions spatial pyramid network for crowd counting and density estimation," *Neurocomputing*, vol. 350, pp. 91–101, 2019.

[21] Z. Zou, Y. Liu, S. Xu et al., "Crowd counting via hierarchical scale recalibration network," 2020, https://arxiv.org/abs/2003.03545.

[22] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, pp. 818–833, Cham, Switzerland, September 2014.

[23] X. Cao, Z. Wang, Y. Zhao et al., "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, Munich, Germany, September 2018.

[24] A. Paszke, S. Gross, S. Chintala et al., *Automatic Differentiation in Pytorch*, in *Proceedings of the NIPS Workshop*, Long Beach, CA, USA, 2017.

[25] V. A. Sindagi and V. M. Patel, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *Proceedings*

of the IEEE International Conference on Computer Vision, pp. 1002–1012, Seoul, Korea, October 2019.

[26] Z. Shen, Y. Xu, B. Ni et al., "Crowd counting via adversarial cross-scale consistency pursuit," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5245–5254, Salt Lake City, UT, USA, June 2018.

[27] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1113–1121, Lake Tahoe, NV, USA, February 2018.