



Dense crowd counting based on adaptive scene division

Ying Yu¹ · Huilin Zhu¹ · Lewei Wang¹ · Witold Pedrycz^{2,3}

Received: 27 March 2020 / Accepted: 20 September 2020 / Published online: 29 September 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

With the rapid development of computer vision and artificial intelligence, crowd counting has attracted significant attention from researchers and many well-known methods were proposed. However, due to interocclusions, perspective distortion, and uneven crowd distribution, crowd counting is still a highly challenging task in crowd analysis. Motivated by granular computing, a novel end-to-end crowd counting network (GrCNet) is proposed to enable the problem of crowd counting to be conceptualized at different levels of granularity, and to map problem into computationally tractable subproblems. It shows that by adaptively dividing the image into granules and then feeding the granules into different counting subnetworks separately, the scale variation range of image is narrowed and the adaptability of counting algorithm to different scenarios is improved. Experiments on four well-known crowd counting benchmark datasets indicate that GrCNet achieves state-of-the-art counting performance and high robustness in dense crowd counting.

Keywords Crowd counting · Granular computing · Density map · Feature extraction · Dilated convolution

1 Introduction

Crowd counting is a fundamental task of crowd analysis. It aims to estimate the number of individuals in a sparse or dense crowd scene. With the rapid urbanization around the worldwide, the urban population is growing rapidly. Exponential growth in the urban population has led to an increased number of activities such as vocal concert, sporting events, political rallies, etc., thereby resulting in more frequent crowd gatherings in the recent years. In such scenarios, it is essential to count the number of individuals in a crowded scene for better management, safety and security [1]. Consequently, crowd counting has emerged as a crucial focus in crowd analysis for providing valuable information to anticipate overcrowding or detect the abnormal events. This endeavour is also further motivated by the need for a sophisticated crowd analysis system.

Crowd counting has a variety of real-world applications, such as public safety management [2, 3], intelligent surveillance [4], and urban planning [5]. The methods developed for crowd counting can be easily extended to object counting tasks in many other domains, such as vehicle counting [6, 7], animal counting [8], etc.

With the rapid development of computer vision and artificial intelligence technology, crowd counting has attracted significant attention from researches in the recent past and many crowd counting algorithms were proposed. In general, the existing crowd counting methods could be categorized into four groups [9]: detection-based methods, clustering-based methods, regression-based methods, and density-estimation-based methods. Among them, neither the detection-based methods nor the clustering-based methods are suitable for handling large-scale and high-density crowd. To address the limitation of detection-based and clustering-based methods, some works used regression-based method that directly learns the mapping from an image patch to the number of crowd. However, the regression-methods only focus on the total number of crowd and cannot provide detailed information, such as the spatial distribution of crowd. The density-estimation-based methods are designed for large-scale object counting. It can estimate the density map of a input image, where each pixel value in the density map corresponds to the crowd density at the corresponding location of the input

✉ Ying Yu
yuyingjx@163.com

¹ College of Software, East China Jiaotong University, Nanchang 330013, China

² Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2G7, Canada

³ System Research Institute, Polish Academy of Sciences, Warsaw, Poland

image. The number of crowd can be obtained by integrating the entire density map. In addition, due to incorporating spatial information in the learning process, the density-estimation-based methods can calculate the number of objects within any region in the density map. Consequently, it is the most popular crowd counting framework right now. In this paper, the density-estimation-based model is employed to predict density map.

Although researchers have attempted to address the crowd counting problem with some success from different viewpoint, the dense crowd counting is still a highly challenging task in computer vision. One of particular challenges is the perspective distortion that results in large variations in size and appearance of objects. Therefore, for the similar objects located in different position of a scene, their features extracted from the image will be different. As shown in Fig. 1 which comes from ShanghaiTech Part A dataset [10], the shape and scale of individual at different location in the image varies due to the camera orientation and position. This can lead to the problem that the extracted features may not offer sufficient discrimination, and thus inevitably lead to incorrect density estimation. Additionally, the ambiguity of features is further improved by varying physical layout of crowd environments. Consequently, in order to overcome the shortcoming of perspective distortion, we need powerful features that have great robustness and adaptability across different scenes. The early work mainly use hand-crafted features extracted from local image to count crowd and the more recent works use Convolutional Neural Network (CNN) [11] based approaches to extract features. The CNN-based approaches have demonstrated significant improvements over previous hand-crafted feature-based methods, motivating more researchers to explore CNN-based approaches further for related crowd counting problems. In



Fig. 1 Dense crowd image

this paper, the CNN-based approach is adopted to extract feature for complex scenes.

The other challenge is diverse crowd densities. As we all know, crowd density varies from regions to regions within a image. For a dense crowd scene, the crowd density of image patch that is further away from the camera appears larger than the crowd density of image patch that is closer to the camera. As illustrated in Fig. 1, the individual standing in the distant region occupies fewer pixels than the individual standing in the nearby region. So even if the image patches are of the same size within a scene, their densities may vary greatly. For the distant region, the frequent, partial or complete occlusion between individuals is more common. It is very difficult to discern individuals in crowd since they are severely interoccluded with each other. Therefore, It is not very appropriate to extract the features of individuals of different region for density regression in a unified method. To solve the problem, some researchers adopted multi-column CNNs architecture to extract multi-scale features. Each column corresponds to filters with receptive fields of different sizes (i.e. small, medium, large) to cope with large variation in individual size due to perspective distortions. Although the multi-column architectures prove the ability to estimate crowd count, several disadvantages also exist in these approaches. They are hard to train caused by the multi-column architecture, and they have large amount of redundant parameters. The computational complexity is also large as multiple CNNs need to be run.

Motivated by the aforementioned shortcomings, we propose a novel end-to-end network called GrCNet based on granular computing for dense crowd counting. In order to increase the robustness and adaptability of features across different scenes, CNN is utilized to automatically extract scene features. Compared with the hand-crafted features, CNN-based features have stronger discriminative ability and is more adaptive for dense crowd counting. In order to reduce the adverse effect of perspective distortion and diverse crowd density distribution, the concept of granular computing is incorporated into the model. Granular computing (GrC) is an emerging computing paradigm of information processing and it simulates human cognitive process by enabling abstraction on the essential details at different granularities [12]. In this paper, we attempt to divide the image into granules at different levels of granularity with the hope that granulation can alleviate the complexity for crowd counting. After adaptively horizontal segmentation, the scene image is divided into two granules, namely distant-shot granule and close-shot granule, according to crowd density level. Each granule consists of many smaller multi-scale individual granules. Then two column crowd counting networks with different receptive field size are employed to capture the multi-scale features of granules. The filter

with larger receptive fields are used for modeling the density maps of the granule composed of larger individuals. Since GrCNet is a density estimation based crowd counting model, a crowd density map is then learned from the multi-scale features through a fully connected network. In this way, a complexity counting problem is mapped into several computationally tractable subproblems. By counting crowd separately, the model's robustness to scale variation can be improved even without perspective distortion correction. The parameters of model and the amount of data required for training are also reduced. Compared with the existing multi-column CNN based methods, GrCNet considers the fact that the density of regions varies within the image. In CrCNet, a complete image is divided into several parts adaptively according to the density level before feature extraction, while in existing multi-column approaches, the image is usually input into the network as a whole. Dividing the scene image first and then sending it to the network can reduce the range of scale variation. Furthermore, compared with the other division based approaches, GrCNet is more flexible. It divides the image adaptively rather than evenly. Additionally, in order to reduce the number of parameters caused by the multi-column network, a dilated convolution is incorporated into the model. The main contributions of this work are as follows:

1. Based on the granular computing, a end-to-end crowd counting network is proposed. It can adaptively divide the image into granules with different density.
2. By narrowing the range of scale variation, the counting performance of proposed network is improved.
3. Dilated convolution is utilized to extract discriminative features while reducing the amount of parameters caused by multi-column CNN.
4. Space pyramid pooling (SPP) is incorporated into the splitting network to ensure that the output is the same size.

The remainder of the paper is organized as follow. Section 2 briefly reviews the related works in crowd counting. Section 3 proposes a framework of crowd counting based on granular computing. This is followed by a detailed description on three main parts of the framework in Sect. 4. They are adaptive distant-close shot Splitting Network (SpliteNet), the Distant-Shot counting Network (DSNet), and the Close-Shot counting Network (CSNet); Sect. 5 conducts experimental comparisons and results analysis of the algorithms proposed in this paper on multiple well-known crowd counting data sets. Finally, concluding remarks are made in Sect. 6.

2 Related works

According to the technology used, the existing crowd counting algorithms can be categorized into two groups: traditional machine learning based methods and CNN-based methods.

2.1 Traditional machine learning based methods

Early works on crowd counting was mainly based on traditional machine learning approaches such as object detection, regression and so on. Detection-based methods first extract the individual's overall features such as Haar wavelets [13], edgelets [14], shapelet [15], and etc., then train a detector to identify individuals and count the number of individuals in a image. However, these methods have obtained limited recognition performance due to the difficult extraction of overall feature caused by the unavoidable occlusion in dense crowd. To overcome this issue, researchers began to consider part-based detection methods. Instead of directly detecting the overall individual, they detect the specific body parts such as the head or shoulder to count individuals [16, 17]. Although the part-based detection methods proposed some solutions to dense crowd counting, the detection-based methods are still only suitable for sparse crowd.

To address the problem of occlusion, several crowd counting methods based on regression [18, 19] are proposed to learn a mapping from the extracted features to the number of individuals. The regression-based method is significantly better than the detection-based method for high-density crowd, but it ignores important spatial distribution information.

Inspired by regression-based methods, researchers proposed several methods based on density estimation [20, 21]. They establish a mapping from the features to the density distribution map, and effectively integrates spatial information into the learning process. After the density distribution map is obtained, the number of people in any area of the image can be counted by integration. Methods based on density estimation are more difficult to implement via traditional machine learning methods, so there are relatively fewer studies.

In short, the crowd counting based on traditional machine learning algorithms generally requires complex processes such as data preprocessing and hand-crafted feature extraction. Due to the potential limitations of algorithm, the counting error increases significantly with the crowd density.

2.2 CNN-based methods

Benefiting from the strong ability of convolutional neural network (CNN) to learn feature representations, a variety of CNN-based crowd counting algorithms have been proposed.

Zhang et al. [22] first introduced CNN into the field of crowd counting, proposed a cross-scene crowd counting

model, and optimized the counting model alternately through density estimation and regional number regression. Subsequently, in order to solve the multi-scale problem of the object, Zhang [10] et al. proposed a multi-column convolutional neural network (MCNN) architecture with several branches for crowd counting. Each branch uses a different-sized convolution kernel to extract multi-scale features of image.

Inspired by MCNN, Sam et al. [23] proposed a new multi-branch switching convolutional neural network (Switch-CNN). It evenly divides the image and adaptively selects the most optimal regressor among several independent regressors for a particular image patch. Instead of combining the features maps together from all branches, Switch-CNN can choose the most appropriate branch according to the density level and uses the features from that branch for density estimation. However, Switch-CNN have bad results when the switch is selected incorrectly.

The counting performance mainly depends on the quality of predicted density map. To improve the quality of the density map of MCNN, Sindagi et al. [24] introduced context information into the crowd counting and developed a contextual pyramid CNN (CP-CNN) that combines both global and local contextual information for achieving low crowd counting error and high-quality density maps. Although the density map is enriched with two additional columns capturing global and local context, CP-CNN suffers from the high computation complexity in predicting the global and local contexts.

To address the problems of ineffective branches and expensive computation existing in the previous multi-column networks. Li et al. [25] proposed a single-column dilated convolutional neural networks called CSRNet [26] for crowd counting. It utilizes the dilated kernels to expand the receptive field while keeping the image size unchanged. In addition, to tackle the varying density and distribution of crowd, PaDNet [27] proposed a new Density-Aware Network (DAN) module to distinguish the variation of crowd density, and a Feature Enhancement Layer (FEL) module to improve global and local recognition performance.

In short, CNN-based method can overcome the shortcoming of hand-crafted features and achieve better counting performance. Even if it suffer from the high computational complexity, it is still the mainstream algorithm in crowd counting.

3 Our approach

Due to the effect of perspective distortion, the density of individuals can vary from region to region and the appearance features of individuals in crowd also have huge diversity, which makes the dense crowd counting problem extremely difficult. To meet it, we propose a dense crowd counting framework GrCNet based on granular computing. It could adaptively divide the scene image according to the density level. The scene image is regarded as a top granule, and then the top granule is divided into two mid-granules with different density. Each of them contains many smaller individual granules. Different mid-granules are fed into different counting channels, and then the discriminative features of granule are learned for density estimation. Relying on the concept of top-down, layer-by-layer decomposition of granular computing, the problem of crowd counting is conceptualized at different levels of granularity, and the influence caused by the perspective distortion is reduced to some extent.

Two key stages of the proposed framework are illustrated in Fig. 2. They are scene adaptive splitting stage and crowd counting stage. The proposed GrCNet consists of three important subnetworks: distant-close shot Splitting Network (SplitNet), Distant-Shot crowd counting Network (DSNet), and Close-Shot crowd counting Network (CSNet). First, a distant-close shot splitting training dataset is created, and each image in the training set is labeled with the distant-close splitting ratio. Then, SplitNet that can adaptively divide scene images is trained on the created dataset. After the adaptive splitting of the scene image, the distant-shot granule is fed into the pre-trained distant-shot crowd counting network DSNet, while the close-shot granule is fed into

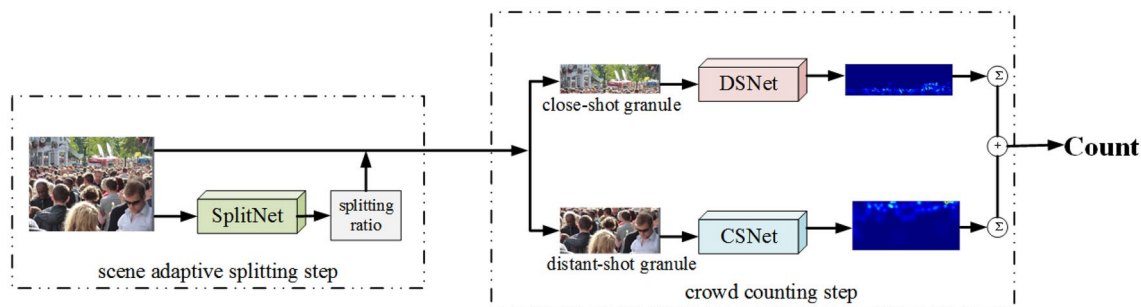


Fig. 2 An illustration of key steps in GrCNet

the pre-trained close-shot crowd counting network CSNet. After extracting the discriminative features from individual granules, two density maps are generated, and the number of individuals can be predicted from the density maps. Following, we will describe GrCNet in details.

3.1 Scene adaptive splitting

The architecture of the scene adaptive splitting network is shown in Fig. 3. First, a distant-close shot splitting network SplitNet is trained, and then the scene image is adaptively divided into two parts: the distant-shot granule and close-shot granule with different density. SplitNet can be regarded as a regression model with a scene image as an input and a splitting ratio as an output.

SplitNet is based on the first ten layers of VGG-16 and then attaches extra five convolution layers to automatically extract the discriminative features of scene image. Spatial pyramid pooling (SPP) is adopted in order to get the same size output from the different size inputs, and uses three different sizes of pooling, namely 1×1 , 2×2 , 3×3 , for the feature maps. The size and stride of the pooling are dynamically adjusted according to the size of feature map so that the

model can deal with input images of any size and maintain a fixed size output. After the data is pooled by SPP, three obtained feature vectors of different size are stitched together and sent to a fully connected layer with only one neuron to obtain an output value that is between 0 and 1. The output value manifests the ratio of the ordinate of splitting point to the height of the image. Finally, the image is divided into two parts according to the splitting ratio, which are called the distant-shot granule and the close-shot granule.

In order to train the model, we need to annotate each image in the training set with a splitting ratio. In dense scenes, most of the individuals in the image are concentrated in the distant region. We divide the distant region and close region according to the number of individuals. The head ordinates are sorted in ascending order to find the ordinate that accounts for r of the total number of individuals, and the ratio t of the found ordinate to the image height is used as the scene splitting point between the distant shot and close shot, where t is between 0 and 1.

After the preparation of dataset is completed, a regression model is trained, where the original scene image is used as input, and the ratio t is used as the output. The training objective is to minimize the gap between the predicted

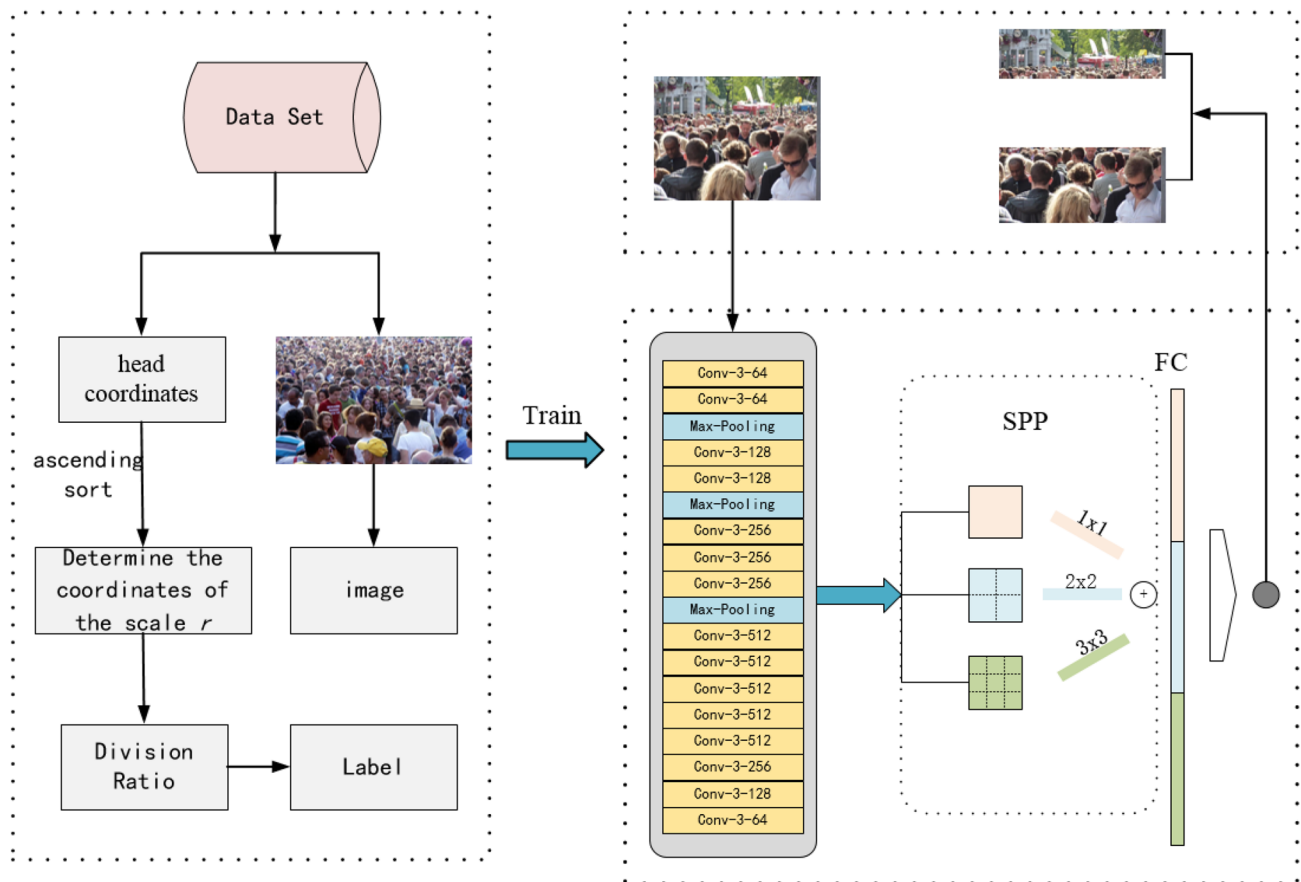


Fig. 3 The architecture of the scene adaptive splitting network

value and the ground-truth value. The Mean Squared Errors (MSE) is utilized as the loss function of the model. The formula reads as:

$$MSE = \frac{1}{N} \sum_{i=1}^N (R_i - R_i^{GT})^2 \quad (1)$$

Among them, N represents the number of samples, R_i represents the predicted splitting ratio, and R_i^{GT} represents the ground-truth splitting ratio.

3.2 Dense crowd counting

SplitNet divides the scene image into two mid-granules: close-shot granule and distant-shot granule according to the density level. In the dense crowd counting stage, two branches of CNN are trained, including distant-shot network (DSNet) and close-shot network (CSNet), to separately extract discriminative features from two mid-granules and then generate density maps for the two mid-granules. Finally, the number of individuals can be estimated from the two density maps. As shown in Fig. 4, it illustrates the process of dense crowd counting stage.

It is widely known that the size of individual in the distant shot is usually smaller than the size of individual in the close shot. Due to this reason, different-sized filters are employed to capture multi-scale information from different regions. A recent study manifests that the filter with larger receptive field is more useful for modeling the density map of region consisting of larger head [27], so a convolution kernel with a larger receptive field is adopted in CSNet. The receptive field is defined as the region of input image that a particular feature is affected by. It determines the affinity towards certain density types. If you want to learn features with larger receptive fields, you need to stack more layers of convolutional operations or larger convolution kernels which greatly increase the amount of network parameters. The dilated convolutional network has proven its effectiveness in pixel-level tasks in multiple fields [25]. It can effectively expand the receptive field without increasing the number of parameters and avoid the loss of spatial information of small feature maps. Therefore, as shown in Fig. 4, CSNet utilizes the first ten layers of VGG-16, then add extra six layers of dilated convolution for feature extraction. Moreover, in order to reduce the amount of parameters, a small convolution kernel is employed.

It has been found through experiments that the dilated convolution works better in dense scenes. Considering that the crowd in the distant is denser than the nearby crowd in a crowded scene, more layers of dilated convolution are used in CSNet. The output of dilated convolution can be expressed as follow:

$$y(m, n) = \sum_{i=1}^M \sum_{j=1}^N x(m + \alpha \times i, n + \alpha \times j) w(i, j) \quad (2)$$

Among them, $y(m, n)$ represents the output of the dilated convolution, $x(m, n)$ is the input, (m, n) represents the shape of the input feature map, α represents the dilated rate, and $w(i, j)$ represents a convolution kernel.

The crowd distribution in the close-shot granule is more sparse than the distant-shot granule. Similar to CSNet, we use the first ten layers of VGG-16 as the base of DSNet, and subsequently append a three-layer dilated convolution. Compared with the DSNet, more layers are more suitable for sparse crowd scenes with larger head.

The counting performance relies on the quality of ground truth. In this work, the ground truth is generated from point annotation available with crowd datasets. The point annotations approximately specify the location of individual's heads, and are also regarded as the center of the Gaussian kernel. Then the density value of each pixel is superposed by the values of multiple corresponding Gaussian kernel function. Only when the ground-truth density map is created can a mapping from the features to the density be established.

Due to suffering from severe perspective distortion, it is not a wise decision to directly perform Gaussian processing on the annotated individual's center to obtain the density map. Therefore, following the method of generating density maps in [10], the geometry-adaptive kernels are adopted to tackle the datasets with crowded scenes. The Gaussian kernel can blur each head label, and the ground truth is also generated from it. This processing can be expressed as:

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

$$\bar{d}_i = \frac{1}{m} \sum_{j=1}^m d_j^i \quad (4)$$

Among them, x_i indicates that there is a head at the x_i pixel, N indicates the number of people in a scene image, \bar{d}_i indicates the average distance between the individual's head and its nearest m neighbors, δ indicates ground truth, and G_{σ_i} indicates that the coordinates of each head are performed Gaussian blurring, σ_i is the standard deviation. We followed the parameter settings of [10], $\beta=0.3$, $m=3$ which gave the best experimental results.

In order to train DSNet and CSNet separately, we divided each image and corresponding ground-truth density map in the dataset into two parts, the distant view and close view. Following the parameter settings of the previous stage, the proportion of individuals in the distant view accounts for r . Then the two parts are fed into the corresponding subnetworks.

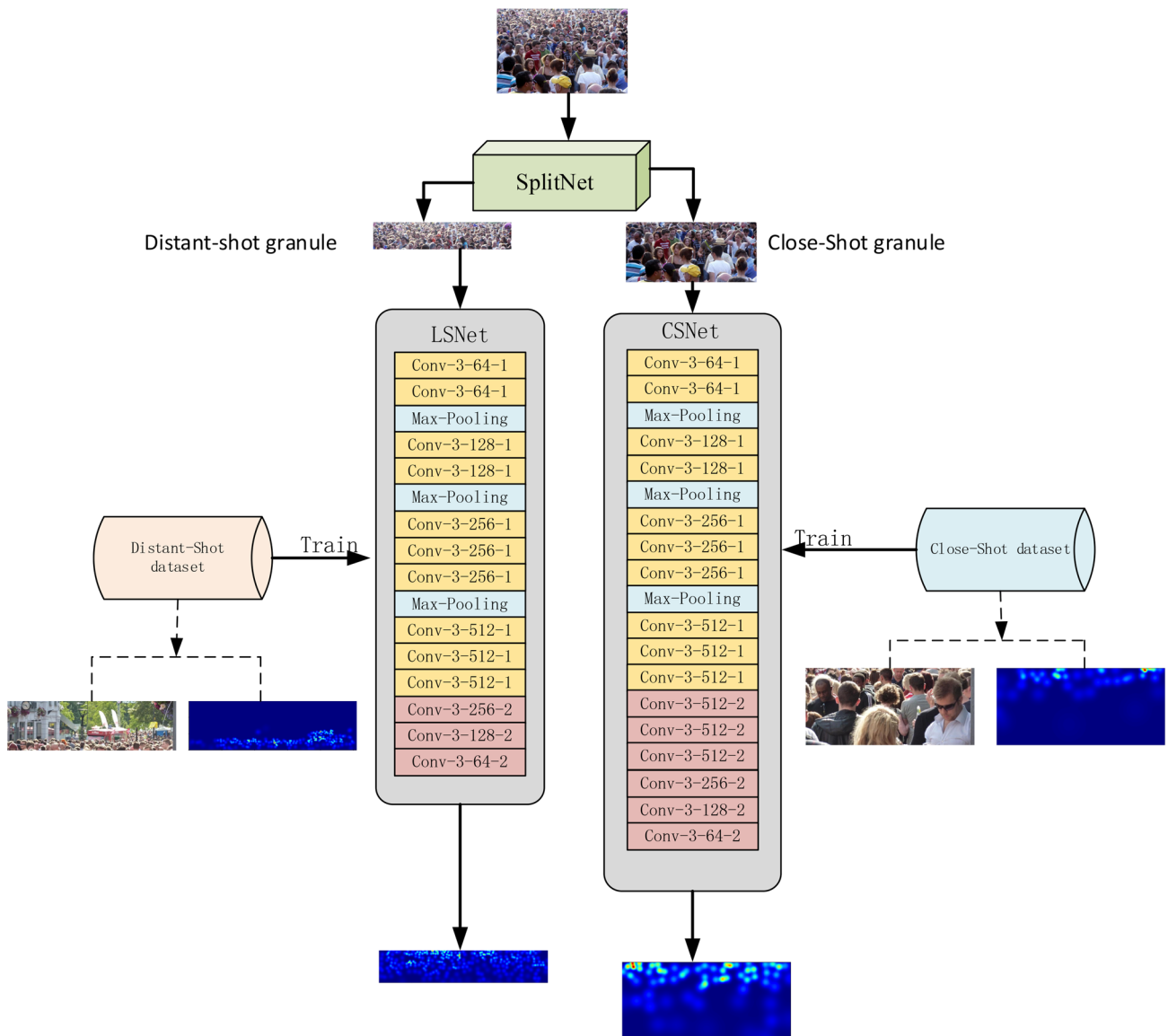


Fig. 4 The framework of dense crowd counting

The target of training is to minimize the gap between the predicted density map and the ground-truth, and the Euclidean distance is employed as the loss function of the two subnetworks. It can be expressed as follow.

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^N \|Z(x_i; \theta) - Z_i^{GT}\|_2^2 \tag{5}$$

Among them, N is the training batch size, $Z(x_i; \theta)$ is the predicted value, Z_i^{GT} is the ground truth value, θ is the training parameters of the network, and x_i is the input image.

Figure 5 shows a example of the density maps separately generated from the close-shot and distant-shot crowd counting networks after adaptively splitting an original image.

As can be seen from Fig. 5, the gap between the predicted density map and the ground truth density map is small.

4 Experimental studies

In this section, we empirically evaluate the proposed algorithm GrCNet by comparing it with other state-of-the-are crowd counting algorithms, including cross-scene crowd counting via deep convolutional neural networks Zhang et al. [22], multi-column convolutional neural network (MCNN) [10], switching convolutional neural network (Switch-CNN) [23], contextual pyramid convolutional neural networks (CP-CNN) [24], congested scene recognition network (CSRNet)

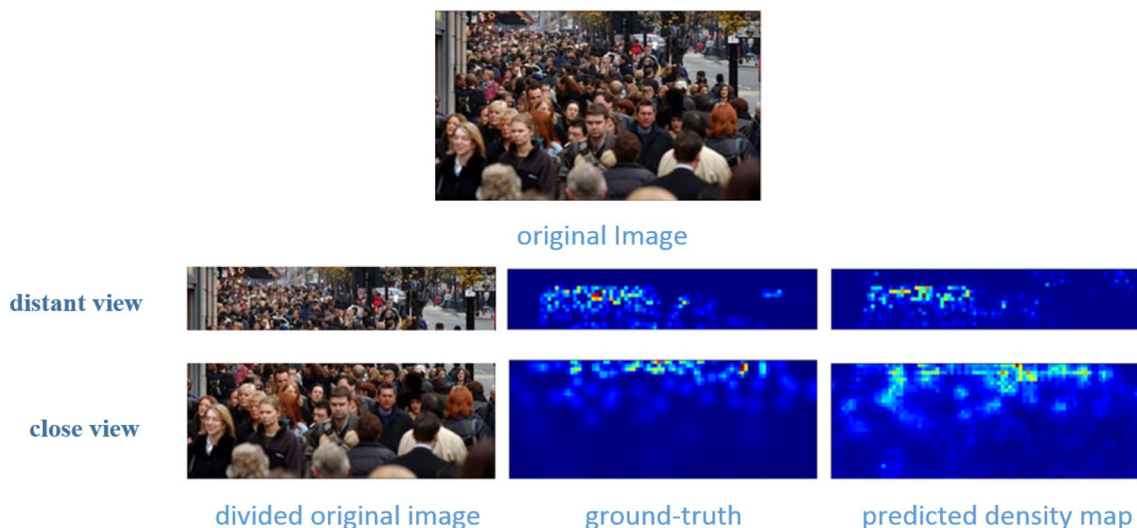


Fig.5 Comparison of predicted density maps with ground truth

[25], iterative crowd counting convolutional neural networks (ic-CNN) [28], and detection and density estimation network (Decidenet) [29]. Furthermore, the influence of dilated ratio on GrCNet is also illustrated. All experiments were run on a server equipped with two E5-2620 V4 processors, 64.0G memory, and two GTX7080TI.

4.1 Datasets

In order to verify the effectiveness of crowd counting algorithm proposed in this paper, we performed experiments on four well-known crowd counting datasets. These datasets differ in the number of samples, resolution, scene type, number of heads, etc., and all provide the true position of the center of object.

1. *Shanghai Tech* [10] this dataset is a benchmark for crowd counting containing 1198 scene images and 330,165 head positions are marked. It is divided into Part A and Part B. Part A, which is densely populated, contains 300 training images and 182 testing images; Part B, which is relatively sparsely populated, contains 400 training images and 316 testing images. Overall, accurate counting on the *Shanghai Tech* is challenging because the data set is diverse in both scene type, perspective and crowd density.
2. *UCF CC 50* [30] this dataset includes 50 images with different resolutions, covering different scenes such as concerts, protests, stadiums and marathons. It is the first truly challenging large-scale crowd counting dataset. A total of 63,075 head positions are marked in the entire dataset. The number of individuals in each image ranges from 94 to 4543, and the density level varies greatly.
3. *WorldExpo'10* [22] it contains 1132 video sequences collected by the 108 cameras from the 2010 Shanghai World Expo. Among them, 3980 frames were annotated manually, the resolution of each frame was 576×720 , and a total of 199,923 object positions were annotated. The dataset is divided into two parts, with 1127 video sequences from 103 scenes as the training set, and the data from the other five scenes as the testing set.
4. *UCSD* [31] it contains 2000 frames of images sampled from a video sequence, each frame has a resolution of 158×238 . Every five frames is labeled manually, and the individual positions in the remaining frames are created using linear interpolation. Finally, 49,885 individuals are labeled. In general, the UCSD dataset is a relatively simple crowd counting data set, because it has a relatively low crowd density, and the scene is also relatively simple.

4.2 Evaluation metrics

This paper adopts the evaluation metrics presented in [10] as the evaluation criteria of crowd counting algorithm, which are the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). MAE is usually used to evaluate the accuracy of model, and RMSE is used to evaluate the robustness of model.

The formulas of evaluation criteria are listed as (6) and (7). N represents the number of images in the dataset, C_i represents the predicted number of individuals, and C_i^{GT} represents the ground-truth number of individuals in the image. C_i can be expressed as Formula (8). Among them, L and W respectively represent the length and width of

the predicted density map, and $z(l, w)$ represents the pixel value of the density map at the point (l, w) .

$$MAE = \frac{1}{N} \sum_{i=1}^N |C_i - C_i^{GT}| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (C_i - C_i^{GT})^2} \tag{7}$$

$$C_i = \sum_{l=1}^L \sum_{w=1}^W z(l, w) \tag{8}$$

In the scene splitting stage, the mean square error (MSE) is used as the evaluation criterion, as shown in (1), where MSE represents the root mean square error between the annotated splitting ratio and the output splitting ratio.

4.3 Results and discussion

Firstly, we empirically study the effect of scene adaptive splitting network. r is set to 0.8, and MSE is used as the loss function. r can also be set to a different value, such as 0.7 or 0.9, but empirical study on the impact of parameter r indicates that the crowd counting can achieve optimum average performance on most datasets when r is equal to 0.8. Figure 6 shows two example of splitting image where r is set to 0.6, 0.7, 0.8, and 0.9 respectively. When r is equal to 0.8, the splitting effect is best.

As can be seen from Fig. 7, MSE shows a downward trend as the iterative number increasing. When the model iterates to 700 steps, it gradually converges and reaches a mean square error of 0.0133. At this time, the image can be accurately divided into close-shot granule and distant-shot granule according to the number of individual. It can be seen from Fig. 6 that two granules of each image are obtained, and the splitting ratios t are equal to 0.32 and 0.44



(a) Original images without splitting

r	0.6	0.7	0.8	0.9
distant-shot				
close-shot				
t	0.13	0.16	0.32	0.45
distant-shot				
close-shot				
t	0.17	0.23	0.44	0.57

(b) Splitting results with different r

Fig. 6 Examples of scene adaptive splitting

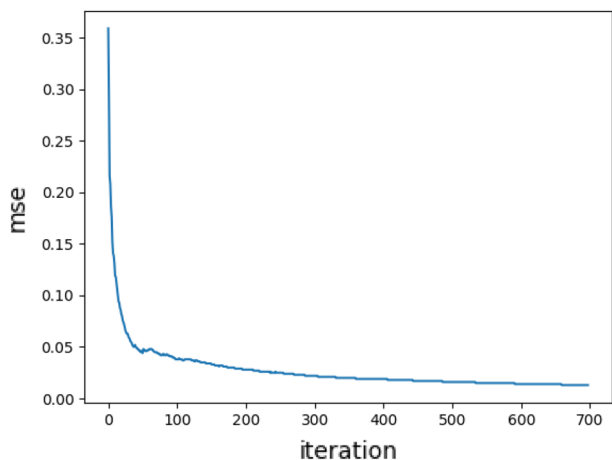


Fig. 7 MSE varies as the number of iteration increasing

Table 1 MAE with different dilated rate

	1	2	3	4
DSNet	57.831	57.261	61.246	60.608
CSNet	12.156	11.539	12.828	12.335

The best values are in bold

respectively when r is set to 0.8. It also can be seen from the splitting results that there are large differences in density, shape, and scale between the nearby objects and the far objects.

Secondly, we empirically compare the impact of the parameters of two crowd counting networks on the *Shanghai Tech* Part A. This includes a comparison of the dilated rates of networks and the number of network layers. The close-shot counting network uses the first ten layers of VGG-16 plus six layers of dilated convolution. Table 1 shows the mean squared error MAE when the dilated ratio is set to 1, 2, 3, and 4 respectively.

As can be seen from the Table 1, the close-shot counting network and the distant-shot counting network both achieved the best results when the dilated rate was set to 2. When the dilated rate was set to 3 or 4, the network works just as well without the dilated convolution.

Table 2 MAE with different layers of dilated convolution

	1	2	3	4	5	6
DSNet	60.082	58.698	57.261	58.192	59.054	57.785
CSNet	13.275	13.561	11.539	12.135	13.280	11.068

The best values are in bold

Table 3 MAE and MSE of different algorithms on *Shanghai Tech*

Model	Part A		Part B	
	MAE	RMSE	MAE	RMSE
Zhang et al. [22]	181.8	277.7	32	49.8
MCNN [10]	110.2	173.2	26.4	41.3
Switch-CNN [23]	90.4	135.0	21.6	33.4
CP-CNN [24]	73.6	106.4	20.1	30.1
CSRNet [25]	68.2	115.0	10.6	16.0
ic-CNN [28]	68.5	116.2	10.7	16.0
GrCNet	67.4	113.2	9.5	14.3

The best values are in bold

After the dilated ratio is fixed at 2, we began to study the impact of the number of network layers. The front-end network uses the first ten layers of VGG-16, and the back-end network respectively uses six kinds of dilated convolution networks with different numbers of layers that is from one-layer to six-layer. For one-layer dilated convolution network, the number of convolution kernels is 64, and the size of the convolution kernel is equal to 3; For two-layer dilated convolution network, the number of convolution kernels is set to 128 and 64 separately, and the size of the convolution kernel is also equal to 3; For three-layer dilated convolution network, the number of convolution kernels is set to 256, 128, and 64 respectively, and the size of the convolution kernel is also set to 3; The four-layer dilated convolution adds a convolution layer with 512 kernels based on the the three-layer architecture, the five-layer dilated convolution network adds a convolution layer with 512 kernels based on the four-layer architecture, and the six-layer dilated convolution network adds two convolution layer with 512 kernels on the basis of four-layer architecture. Table 2 shows the MAEs of these six kinds of network.

It can be seen from Table 2 that the distant-shot counting network achieved best result when it has three layers of dilated convolution, where MAE achieves 57.261. As for close-shot counting network, it gets the best performance when it has six layers of dilated convolution, where MAE is 11.068.

Finally, we compare our algorithm with various state-of-the-art crowd counting algorithms on four benchmark

datasets, including MCNN, Switch-CNN, CP-CNN, CSR-Net, and et al.. Results in terms of *MAE* and *MSE* are shown in Tables 3–6.

Table 3 shows the experimental results on the *Shanghai Tech* dataset, and optimal results are shown in bold. As can be seen from it, compared with some previous algorithms, the counting accuracy of GrCNet is improved. Compared with Part A, the experimental results on Part B is better. In Part B, the *MAE* of GrCNet dropped to 9.5 and the *MSE* of GrCNet dropped to 14.3. It is obviously better than other algorithms. In Part A, the *MAE* of GrCNet is equal to 67.4 and the *MSE* of GrCNet is equal to 113.2. Compared with other algorithms, although *MSE* is sub-quality, *MAE* remains optimal.

Table 4 is the experimental results on the *UCF CC 50* dataset. As can be seen from it, the *MAE* of GrCNet is lower than other algorithms, and the *MSE* of GrCNet also performs well, second only to the *MSE* of CP-CNN.

Table 5 is the experimental results on the *WorldExpo'10* dataset. There are five different scenarios in this data set, which are represented by S1, S2, S3, S4 and S5. As can be seen from Table 5, in scenario 2, scenario 3, and scenario 5, GrCNet achieved good results, and obtained *MAE* of 10.8, 8.4, and 2.8 respectively. Although in the other two scenarios GrCNet did not achieve the optimal result,

Table 4 MAE and MSE of different algorithms on *UCFF CC 50*

Model	MAE	RMSE
Idrees et al. [30]	419.5	541.6
Zhang et al	467.0	498.5
MCNN	377.6	509.1
Switch-CNN	318.1	439.2
CP-CNN	295.8	320.9
CSRNet	266.1	397.5
ic-CNN	260.9	365.5
GrCNet	259.2	336.6

The best values are in bold

Table 5 MAE of different models on WorldExpo'10

Model	S1	S2	S3	S4	S5	Average
Zhang et al	9.8	14.1	14.3	22.2	3.7	12.9
MCNN	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN	2.9	14.7	10.5	10.4	5.8	8.9
CSRNet	2.9	11.5	8.6	16.6	3.4	8.6
ic-CNN	17.0	12.3	9.2	8.1	4.7	10.3
DecideNet [29]	2.0	13.14	8.9	17.4	4.75	9.23
GrCNet	2.2	10.8	8.4	12.2	2.8	7.3

The best values are in bold

Table 6 MAE and MSE of different models on UCSD

Model	MAE	RMSE
Zhang et al	1.60	3.31
MCNN	1.07	1.35
Switch-CNN	1.62	2.10
CSRNet	1.16	1.47
GrCNet	0.96	1.12

The best values are in bold

they are very close to the optimal result, and the average is optimal.

Table 6 is the experimental results on the *UCSD* dataset. As can be seen from the table, our algorithm achieved optimal results both on *MAE* and *MSE*.

In summary, the above experiments indicate that GrCNet has better advantages in terms of accuracy and robustness. In addition, compared with the previous multi-branch structure or dilated convolution structure, GrCNet can reduce the network parameters. These experimental results also demonstrate that it is feasible to adopt the idea of granular computing to divide the image into granules first and then perform the crowd counting problem.

5 Conclusions

This paper presented a novel end-to-end network GrCNet for crowd counting. Motivated by the advantages of granularity computing [32], a scene image is adaptively divided into close-shot granule and distant-shot granule according to the density level, and then sent to different counting network for density maps. By reducing the scale variation range of object, our network obtains predictive performance compared to other state-of-the-art approaches while maintaining fewer network parameters. Future research could further explore the subdivision method of scene density levels, and build a more detailed multi-level granulation structure to improve the counting performance.

Acknowledgements The authors would like to thank the editors for their kindly help and the anonymous referees for their valuable comments and helpful suggestions. The work is partially supported by the National Natural Science Foundation of China (Serial No. 61563016, 61762036), the Natural Science Foundation of Jiangxi Provincial (Serial No. 20181BAB202023, 20171BAB202012).

References

- Sindagi VA, Patel VM (2018) A survey of recent advances in CNN-based single image crowd counting and density estimation. *Pattern Recognit Lett* 107:3–16
- Zhou B, Tang X, Wang X (2015) Learning collective crowd behaviors with dynamic pedestrian-agents. *Int J Comput Vis* 111(1):50–68
- Huang L, Chen T, Wang Y et al (2015) Congestion detection of pedestrians using the velocity entropy: a case study of Love Parade 2010 disaster. *Phys A* 440:200–209
- Qin XH, Wang XF, Zhou X et al (2013) Crowd count in a variety of crowd density scenarios. *J Image Graphics* 04:37–43 (in Chinese)
- Sam DB, Surya S, Babu RV, et al (2017) Switching convolutional neural network for crowd counting. In: *IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 2017, pp 4031–4039
- Zhang H, Kyaw Z, Chang S, et al (2017) Visual translation embedding network for visual relation detection. In: *IEEE Conference on computer vision and pattern recognition*, Honolulu, USA, 2017, pp 3107–3115
- Zhang S, Wu G, Costeiraz JP, et al (2017) FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras. In: *2017 IEEE International Conference on Computer Vision (ICCV) Venice, Italy*. IEEE, 2017, pp 3687–3696
- Arteta C, Lempitsky V, Zisserman A, et al (2016) Counting in the Wild. In: *European Conference on computer vision*, Amsterdam, The Netherlands, 2016, pp 483–498
- Liu X (2018) Research on target counting method in video surveillance. Doctoral thesis, University of Science and Technology of China, Hefei (in Chinese)
- Zhang Y, Zhou D, Chen S, et al (2016) Single-image crowd counting via multi-column convolutional neural network. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Las Vegas, USA: IEEE, 2016, pp 589–597
- Lecun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
- Kok VJ, Chan CS (2017) GrCS: granular computing-based crowd segmentation. *IEEE Trans Cybern* 47(5):1157–1168
- Viola P, Jones MJ (2004) Robust real-time face detection. *Int J Comput Vis* 57(2):137–154
- 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*. Beijing, China, 2005, pp 90–97
- Sabzmeydani P, Mori G (2007) Detecting pedestrians by learning shapelet features. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Minneapolis: IEEE, 2007, pp 1–8
- Felzenszwalb PF, Girshick RB, McAllester D et al (2010) Object detection with discriminatively trained part-based models. *IEEE Trans Pattern Anal Mach Intell* 32(9):1627–1645
- Li M, Zhang Z, Huang K, et al (2008) Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: *International Conference on pattern recognition*. Tampa, USA, 2008, pp 1–4
- Chan AB, Vasconcelos N (2009) Bayesian poisson regression for crowd counting. In: *IEEE 12th International Conference on computer vision*. Kyoto, Japan: IEEE, 2009, pp 545–551
- Ryan D, Denman S, Fookes CB, et al (2009) Crowd counting using multiple local features. In: *Proceeding of digital image computing: techniques and applications*. Melbourne, Australia: IEEE, 2009, pp 81–88
- Lempitsky V, Zisserman A (2010) Learning to count objects in images. In: *Advances in neural information processing systems*, Vancouver, Canada, 2010, pp 1324–1332
- Pham VQ, Kozakaya T, Yamaguchi O, et al (2015) COUNT Forest: CO-voting uncertain number of targets using random forest for crowd density estimation. In: *International Conference on computer vision (ICCV 2015)*. Santiago, Chile: IEEE, 2015, pp 3253–3261.
- Zhang C, Li H, Wang X, et al (2015) Cross-scene crowd counting via deep convolutional neural networks. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Boston, USA: IEEE, 2015, pp 833–841
- Sam, DB, Surya S, Babu RV (2017) Switching convolutional neural network for crowd counting. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Honolulu, USA: IEEE, 2017, pp 4031–4039
- 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 (ICCV)*. Venice, Italy: IEEE, 2017, pp 1879–1888
- Li Y, Zhang X, Chen D (2019) CSRNet: dilated convolutional neural networks for understanding the highly congested scenes. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Salt Lake City, USA: IEEE, 2018, pp 1091–1100.
- Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. In: *4th International Conference on learning representations*. San Juan, Puerto Rico, : Arxiv, 2015: 1511.07122
- Tian Y, Lei Y, Zhang J et al (2020) PaDNet: pan-density crowd counting. *IEEE Trans Image Process* 29:2714–2727
- Ranjan V, Le H, Hoai M (2018) Iterative crowd counting. In: *Proceedings of the European Conference on computer vision*. Munich, Germany : IEEE, 2018, pp 270–285
- Liu J, Gao C, Meng D, et al (2018) Decidenet: counting varying density crowds through attention guided detection and density estimation. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Salt Lake City, USA: IEEE, 2018, pp 5197–5206
- Idrees H, Saleemi I, Seibert C, et al (2013) Multi-source multi-scale counting in extremely dense crowd images. *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Portland, USA: IEEE, 2013, pp 2547–2554
- Chan AB, Liang ZSJ, Vasconcelos N (2008) Privacy preserving crowd monitoring: counting people without people models or tracking. *Proceedings of the IEEE Conference on computer vision and pattern recognition*. Anchorage, USA: IEEE, 2008, pp 1–7
- Gao C, Zhou J, Miao D, Wen J, Yue X (2020) Three-way decision with co-training for partially labeled data. *Inf Sci*. <https://doi.org/10.1016/j.ins.2020.08.104>

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