IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Crowd counting by the dual-branch scale-aware network with ranking loss constraints

ISSN 1751-9632
Received on 1st September 2019
Revised 9th December 2019
Accepted on 20th January 2020
E-First on 20th February 2020
doi: 10.1049/iet-cvi.2019.0704
www.ietdl.org

Qin Wu[1,2] ✉, *Fangfang Yan*[1], *Zhilei Chai*[1,2], *Guodong Guo*[3]

[1]*Department of Computer Science, Jiangnan University, Wuxi 214122, People's Republic of China*
[2]*Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, People's Republic of China*
[3]*Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26505, USA*
✉ *E-mail: qinwu@jiangnan.edu.cn*

Abstract: Image crowd counting is a challenging problem. This study proposes a new deep learning method that estimates crowd counting for the congested scene. The proposed network is composed of two major components: the first ten layers of VGG16 are used as the backbone network, and a dual-branch (named as Branch_S and Branch_D) network is proposed to be the second part of the network. Branch_S extracts low-level information (head blob) through a shallow fully convolutional network and Branch_D uses a deep fully convolutional network to extract high-level context features (faces and body). Features learnt from the two different branches can handle the problem of scale variation due to perspective effects and image size differences. Features of different scales extracted from the two branches are fused to generate predicted density map. On the basis of the fact that an original graph must contain more or equal number of persons than any of its sub-images, a ranking loss function utilising the constraint relationship inside an image is proposed. Moreover, the ranking loss is combined with Euclidean loss as the final loss function. Our approach is evaluated on three benchmark datasets, and better results are achieved compared with the state-of-the-art works.

1 Introduction

With the explosion of world population, there are more and more large-scale population gathering scenes such as holiday trips, sports events, and political gatherings. In recent years, stampede incidents have occurred frequently and caused a large number of casualties. Data analysis shows that overcrowding of a large number of people lacking effective management and control is the main cause of the accidents. If population information could be obtained and analysed in time, then some safety precautions could be taken to avoid the occurrence of tragedies [1]. In addition, crowd information is of great significance for many industries. In public buildings such as railway stations and shopping centres, analytical data of crowds can be used to rationally design public spaces and optimise the safety of public spaces from the perspective of crowd safety and convenience [2, 3]. Crowd counting can also be used to collect information for further analysis and reasoning [4]. For example, supermarkets can optimise the number of employees based on the flow of people at a different time of day. The widespread use of crowd information has made the problem of crowd counting become a hot issue in current research.

Fig. 1 *Samples in the ShanghaiTech part_A dataset [14]. The scale varies significantly within the scene and between scene*

Like other visual tasks, estimation of a number of people and density of population distribution based on images is challenging. Many problems need to be solved such as occlusion, uneven distribution of pedestrians, uneven illumination, different scene, different scale difference, and perspective variation. In recent years, the complexity of this problem and the widespread use of crowd analysis have attracted more and more attention to researchers. Some early methods solved the crowd counting problem by detecting pedestrian in images [5, 6]. Sliding window detectors were used to detect a pedestrian in an image and a corresponding number of people are counted. However, severe occlusion makes these methods perform poorly in crowded scenes. To overcome the occlusion problem, regression-based methods [7, 8] were proposed. First, a variety of manual features such as Haar wavelets [9] and histogram of oriented gradients [10] were extracted for generating low-level information, then a regression model was learnt to convert the counting problem into a regression problem. However, it is still difficult for regression-based approaches to deal with high-density crowd scenarios. Owing to the powerful learning capabilities of neural networks, recent works [11–13] started to use convolutional neural networks (CNNs) to solve the crowd counting problems, and the results were significantly improved. To accommodate variation of head sizes due to perspective effects or image resolutions, as shown in Fig. 1, which significantly influences the performance, some researchers designed multi-column [15, 16] or multi-resolution [17] network structures. Zhang *et al.* [14] proposed a multi-column CNN, which was able to process images of any sizes. The network consists of three sub-networks with different kernel sizes in different convolution layers to handle targets with different scales. Moreover, the final feature is obtained by fusing extracted features from the three sub-networks. This design increases the ability of the network to handle scale variations. To deal with scale variations, Sam *et al.* [18] proposed the Switch-CNN, which used a density classifier to select different regressions for a particular

input patch. Although approaches use multi-column architectures (multi-column convolutional neural network (MCNN)) or density classifiers achieve the most advanced performance so far, a lot of computational effort is needed for regressions. In this paper, we propose a method that combines two network branches with different depths to extract features of different scales, and these features are fused to get a predicted density map for the counting task. Furthermore, by exploiting the counting relationship inside an image, we design a ranking loss function as a part of the final loss function to improve the prediction accuracy.

In summary, the main contribution of this paper is as follows:

(1) A new learning framework for crowd density estimation and crowd counting with a dual-branch scale-aware network (DBSAN) is proposed. Branches with different depths are used to extract information on different scales.
(2) Dilated convolution is applied to generate high-resolution density map for accurate crowd counting with less computation comparing with normal convolution.
(3) On the basis of the fact that number of people in a crowd image are no less than the number of people in any of its sub-images cropped from the original one, we propose a novel training loss function named ranking loss. Moreover, it is combined with traditional loss function as a final loss to make the network converge faster and have better performance.

Our approach is evaluated on three benchmark datasets (shanghaiTech, UCF_CC_50, and UCF_QNRF), and it outperforms the state-of-the-art approaches.

2 Related work

In recent years, the improvement of computing power by graphics processing unit and the emergence of many large datasets have prompted deep learning to achieve excellent performance in many computer vision fields. Inspired by the success of CNNs in image classification [19, 20] and object detection [21–23], researchers began to apply deep learning methods in the field of crowd counting [24–26]. Moreover, CNNs were applied to learn a non-linear mapping from a crowd image to its corresponding density map. This has made a great improvement in the accuracy of crowd counting, especially in high-density crowd scenarios.

Wang *et al.* [27] proposed an end-to-end CNN regression model for high-density crowds. They used a well known network architecture called AlexNet [28], and they added some background pictures without people as negative samples into training data to reduce the impact of background noise. Boominathan *et al.* [29] combined a deep network and shallow network into a two-column fully CNN, and such a network fusion can effectively extract high-level context features and low-level spatial structural features, which helps the crowd counting in the case of large-scale changes. The Switch-CNN proposed by Sam *et al.* [18] used a density classifier to select different network branches for each input patches. They used pre-trained VGG16 to classify density levels of crowd images and performed training according to the classification results. However, in real-time crowded scene analysis, it is difficult to determine the granularity of the density level due to the large variation in the number of people. If we chose fine-grained classifiers, then more columns should be implemented, which is more complex and leads to more redundancy.

Taking the above shortcomings into account, Li *et al.* [30] proposed CRSNet, and they chose a simple model of a single-column network. In CRSNet, the authors adjusted the network to make better use of VGG16 to generate accurate density maps. They designed a back-end structure, where dilated convolution was introduced. It reduces the loss of spatial information and generates a density map with higher resolution. Idrees *et al.* [31] proposed a method to solve crowd counting, density map estimation, and pedestrian localisation in dense crowd images simultaneously based on the fact that the three problems are inherently related.

To solve the problem of a limited number of labelled crowd counting images, Liu *et al.* [17] proposed a novel crowd counting method based on the observation that an original crowded scene image contains no fewer people than its sub-image, which made use of a large number of unmarked crowd images.

3 DBSAN with ranking loss constraints

3.1 Structure of DBSAN

In this section, we first introduce the proposed network structure, which is named as DBSAN. The overall framework is shown in Fig. 2. The basic idea of this paper is to obtain different scale information from different branches and add constraints to the loss function to make the system have better learning ability.

As we all know, crowd images captured from different views result in scale variations of heads. Usually, people near the camera get clear detail information, their faces and body are captured and the scale of the head is large. Contrarily, if a person was away from the camera or captured from an aerial viewpoint, then only rough information of the person is captured, the person appears as a head blob in the image and the scale of the head is small. On the basis of this observation, we design a network, which has two different branches to get information for people of different scales.

We choose the first ten layers of VGG16 [32] as the backbone network (marked as blue in Fig. 2) because of its strong migration learning ability and flexible architecture.

For small size targets, deeper network structures will lose more information. To extract features of different scales, we design two different network branches as the second part of our network (marked as green in Fig. 2). Information on different scales is extracted from the two branches. The shallow network (denoted as Branch_S (BS)) is better for extracting lower-level information (head blob). The deeper network (denoted as Branch_D (BD)) is better for extracting high-level context features (faces and body). Feature maps from different depths capture information of different scales. This is the key to solve the problem of different scales caused by different perspectives.

Meanwhile, to reduce the loss of spatial information, the second part of the network uses dilated convolution instead of conventional convolution. The details of dilated convolution are introduced in Section 3.2.

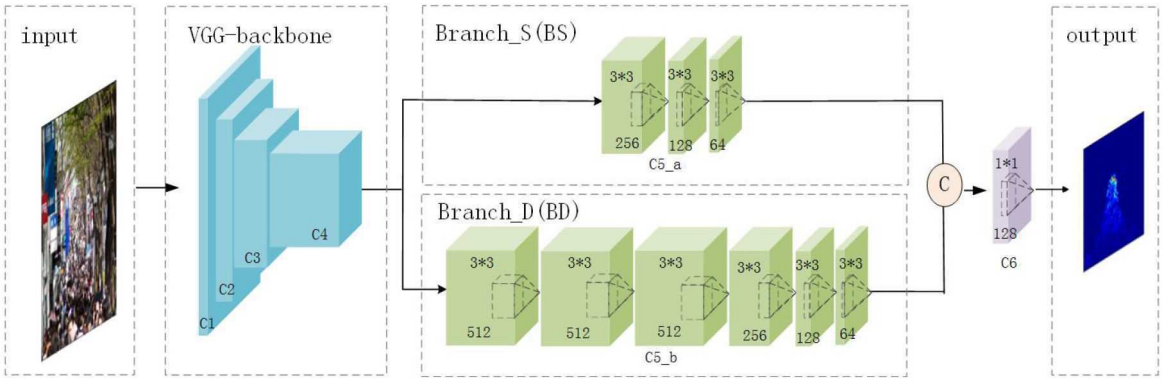

Fig. 2 *Architecture of the proposed DBSAN*

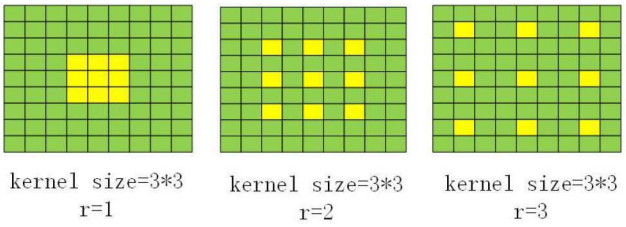

Fig. 3 3×3 *Convolution kernels with different dilation rates*

Table 1 Configuration of convolution layers in the proposed DBSAN

Backbone	Block	Layers	
VGG-backbone	C1	C-3*3-64	
		C-3*3-64	
		MP-2*2	
	C2	C-3*3-128	
		C-3*3-128	
		MP-2*2	
	C3	C-3*3-256	
		C-3*3-256	
		C-3*3-256	
		MP-2*2	
	C4	C-3*3-512	
		C-3*3-512	
		C-3*3-512	
		MP-2*2	
dual branch	C5	**Branch_S (C5_a)**	**Branch_D (C5_b)**
		D-3*3-256-2	D-3*3-512-2
		D-3*3-128-2	D-3*3-512-2
		D-3*3-64-2	D-3*3-512-2
			D-3*3-256-2
			D-3*3-128-2
			D-3*3-64-2
fusion	C6	C-1*1-1	

Fig. 4 *Ranked sub-images*

Finally, features extracted from the two branches are fused to generate a density map. To get a better estimation of crowd number in a crowd image, we also propose a ranking loss function based on the constraint between an image and its sub-image, which is combined with the traditional Euclidean loss as the final loss.

3.2 Dilated convolution

In normal convolutional networks, convolution layers are usually followed by pooling layers. However, pooling layers have the following problems: pooling layers are unlearnable, internal data structure, and spatial–hierarchical information are lost, small object information cannot be reconstructed. To overcome these problems,

Input: A crowd scence image I,
number of subimages K,
scale factor α.
Output: subimages $I_1, ..., I_K$.
(x, y) = center coordinates of I.
W = width of I,
H = height of I,
$I_1 = I$
for each $i \in [2, K]$ **do**
$W_i = W \times \alpha^{i-1}$,
$H_i = H \times \alpha^{i-1}$,
get subimage I_i by cropping image I with center (x, y), width W_i and height H_i.
resize I_i to the size of image I.
end for

Fig. 5 *Algorithm 1: generate ranked sub-images*

dilated convolutions are applied to the second part (both of Branch_S and Branch_D) of our network.

Compared to normal convolutions, dilated convolutions inject holes into convolution kernels to increase receptive fields. Dilated convolution was first introduced in segmentation tasks, the purpose of its structure is to provide a larger receptive field without using pooling layer, and with the same amount of computation compared with normal convolution.

A two-dimensional dilated convolution is defined in the equation below:

$$q(i,\ j) = \sum_{s=1}^{S} \sum_{t=1}^{T} p(i + r*s,\ j + r*t) w(s,\ t) \tag{1}$$

where $p(i, j)$ is the value at a location (i, j) in the input feature map, $q(i, j)$ is the corresponding output, $w(s, t)$ is a convolution kernel with width S and height T, and the parameter r is the dilation rate.

As shown in Fig. 3, with a kernel size of 3*3, a dilated convolution with $r = 1$ turns into a normal convolution corresponding to a 3*3 receptive field, a dilated convolution with $r = 2$ corresponds to a 5*5 receptive field, and a dilated convolution with $r = 3$ corresponds to a 7*7 receptive field.

The configurations of convolution layers in the proposed DBSAN is shown in Table 1. Configurations of normal convolutional layers are denoted as 'C−(kernel size)−(number of filters)', configurations of dilated convolutional layers are denoted as 'D−(kernel size)−(number of filters)−(dilation rates)', and the size of feature map is the same before and after convolution operation. Configurations of max-pooling layers are denoted as 'max-pooling (MP)-(kernel size)'. As shown in Table 1, Branch_S consists of three 3*3 dilated convolutional layers and Branch_D consists of six 3*3 dilated convolution layers. Moreover, feature maps of Branch_S and Branch_D are finally fused by a 1*1 convolution.

The dilated convolution has the characteristics of retaining internal data structure and avoiding downsampling, so it is a better choice for a large receptive field.

3.3 Loss function

Generally, crowd counting methods based on density estimation mainly use Euclidean distance between predicted density map and ground-truth (GT) density map as the loss function. We name it as Euclidean loss (LE) and it is defined below:

$$\mathrm{LE} = \frac{1}{M} \sum_{l=1}^{M} \| \hat{D}(X_l) - D(X_l) \|_2^2 \tag{2}$$

where LE is the Euclidean loss, M is the number of images in a training batch, $\hat{D}(X_l)$ is the density map of the lth image in the training batch, and $D(X_l)$ is the GT density map of the lth image.

As mentioned in SANet [33], Euclidean loss assumed pixels and pixels are independent. It ignores the internal relationship

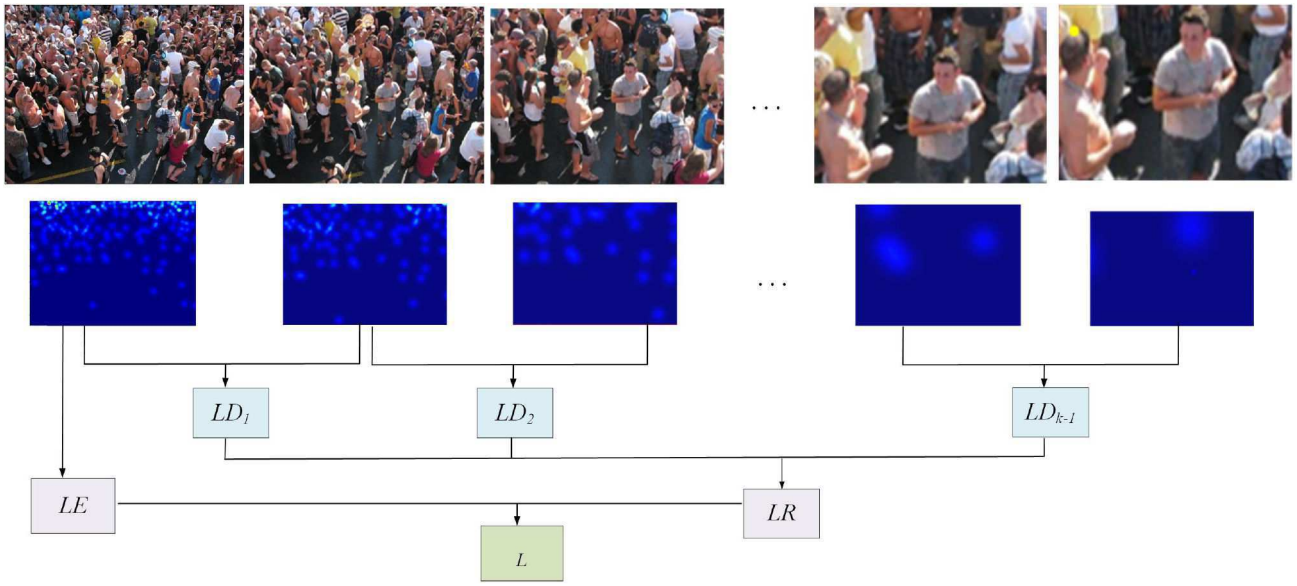

Fig. 6 *Loss function*

inside an image. To solve this problem, we propose a ranking loss, which uses the constraint of the number of relationships between an image and its sub-image.

For any crowd image, it contains no fewer people than any of its sub-images. As shown in Fig. 4, we crop the original image in descending order to get sub-images $I_1, I_2, \ldots, I_{K-1}, I_K$ and $I_K \subseteq I_{K-1} \subseteq \cdots \subseteq I_2 \subseteq I_1$. From Fig. 4, one may easily find that counts of the crowd in any sub-image I_i [denoted as $C(I_i)$] must satisfy the following conditions:

$$C(I_K) \leq C(I_{K-1}) \leq \cdots \leq C(I_2) \leq C(I_1) \tag{3}$$

On the basis of this observation, we proposed a new loss function, named as a *ranking loss*. Before we give the formal definition of the ranking loss function, we first introduce the way we generate ranked sub-images. It is shown in Algorithm 1 (see Fig. 5).

Assume that the predicted number of people in the image I_i is $\hat{C}(I_i)$, then we should have

$$\hat{C}(I_K) \leq \hat{C}(I_{K-1}) \leq \cdots \leq \hat{C}(I_2) \leq \hat{C}(I_1) \tag{4}$$

where $\hat{C}(I_k)$ is calculated by the formula below:

$$\hat{C}(I_k) = \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} \hat{d}_k(i, j) \tag{5}$$

where $\hat{d}_k(i, j)$ is the value of density at a location (i, j) in the predicted density map of sub-image I_k.

For any image I, we define the ranking difference between its sub-image k and sub-image $k+1$ as $\mathrm{LD}(I, k)$, which is calculated by the formula below:

$$\mathrm{LD}(I, k) = \max\left(0, (\hat{C}(I_k) - \hat{C}(I_{k+1}))\right) \tag{6}$$

Moreover, we define the ranking loss of a single image I and its sub-images as $\mathrm{LRS}(I)$, which is calculated by the formula below:

$$\mathrm{LRS}(I) = \sum_{k=1}^{K-1} \mathrm{LD}(I, k) \tag{7}$$

where K is the number of sub-images.

The ranking loss of the network, denoted as ranking loss (LR), is defined in the formula below:

$$\mathrm{LR} = \frac{1}{M} \sum_{l=1}^{M} \mathrm{LRS}(X_l) \tag{8}$$

where M is the number of training batch.

As shown in Fig. 6, the final loss function of our network is a combination of two losses: the Euclidean loss (LE) used in most of the related work and the ranking loss (LR) we proposed. The final loss function of the network is formulated as a weighted sum of these two losses, as the formula below:

$$L = \mathrm{LE} + \lambda \mathrm{LR} \tag{9}$$

where λ is a parameter to balance contributions of Euclidean loss and ranking loss, the selection of λ will be described in Section 4.3.1.

3.4 GT density map

Existing crowd counting datasets only provide coordinates of positions of human heads in images. In our approach, a predicted density map is calculated to approximate the GT density map, and then we use the predicted density map to calculate the estimated counting. To get the GT density map of crowd counting datasets, we use the method proposed in MCNN [14], which uses a geometric adaptive kernel function. We first represent a head labelled at pixel z_i as a delta function $\delta(z - z_i)$. $\delta(z - z_i) = 1$ if there is a head of a pedestrian at position z_i and $\delta(z - z_i) = 0$ otherwise. Thus, an image with V labelled heads can be represented as shown below:

$$H(z) = \sum_{i=1}^{V} \delta(z - z_i) \tag{10}$$

To convert an image to a continuous density map, we convolve $\delta(z - z_i)$ with G_{u,σ^2} to generate the density map, and the corresponding GT density map is calculated by the equation below:

$$D(z) = \sum_{i=1}^{V} \delta(z - z_i) * G_{u,\sigma^2}(z) \tag{11}$$

where V is the number of people in the crowded image, z_i represents the location of the ith head in the image, $G_{u,\sigma^2}(\cdot)$ represents a Gaussian kernel with mean u and variance σ^2.

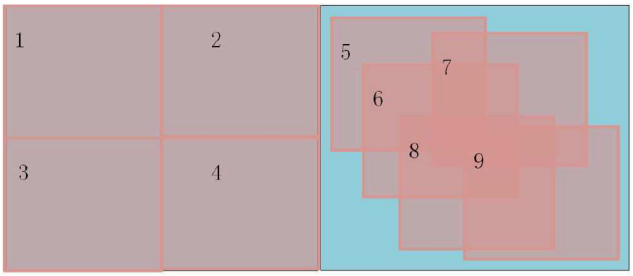

Fig. 7 *Cropping method of nine patches*

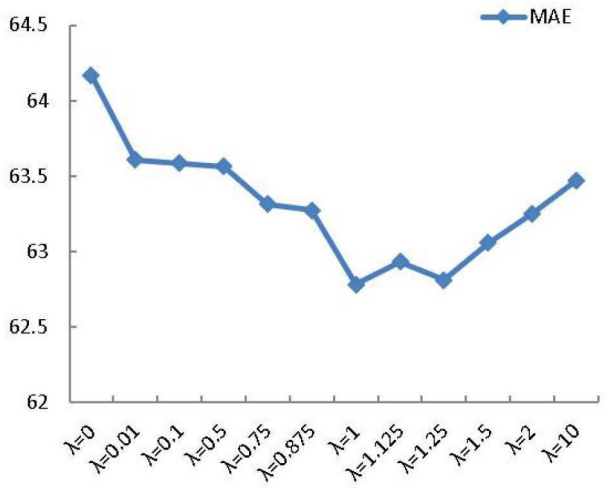

Fig. 8 *MAE with different values of lambda*

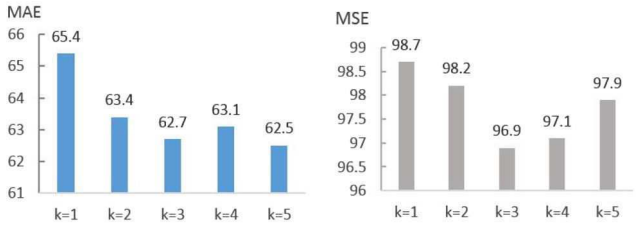

Fig. 9 *Result of different numbers of sub-images k*

4 Experiment

4.1 Datasets

In this paper, we choose three commonly used datasets to evaluate our method:

1. ShanghaiTech [14]: The ShanghaiTech dataset contains 1198 annotated images with a total of 330,165 heads. This dataset is divided into two parts, part_A with highly congested scenes and part_B with relatively sparse crowd scenes. part_A contains 482 images (300 training images and 182 test images) and part_B contains 716 images (400 training images and 316 test images).
2. UCF_CC_50 [34]: The UCF_CC_50 dataset contained 50 images with different sizes. Moreover, the number of people per picture ranges from 94 to 4543. We adopt the five-fold cross-validation method, as mentioned in [34] for this dataset.
3. UCF_QNRF [31]: The UCF_QNRF dataset is a new, large dataset for evaluating crowd counting and localisation. It contains 1535 pictures of dense crowds including 1201 training images and 334 testing images. The UCF_QNRF dataset owns the largest number of high-density crowd images and annotations, which makes this dataset very complex.

4.2 Training details

Data augmentation: To get more training images, image augmentation is applied at the training stage. Each training image is cropped by different ways to get nine patches first. Moreover, the size of each patch is 1/4 of its corresponding original images. The cropping method is shown in Fig. 7. The first four patches are obtained by dividing the original image into four equal size sub-images without overlapping. The other five patches are obtained by randomly cropping the image with different centre locations. Moreover, then we select one patch from the nine patches and put its mirror image into the training set. Thus, the number of training images is doubled.

Parameter initialisation: Parameters of the VGG-backbone network are initialised by the pre-trained VGG net on ImageNet. The remaining parameters of the network are initialised with Gaussian distribution with mean 0 and variance 0.01. We train the network using a stochastic gradient descent optimiser with momentum set to 0.95. The initial learning rate is set to 1×10^{-7}, and it is adaptively reduced according to the number of iterations. In our experiments, we empirically set the scale factor of sub-images as $\alpha = 0.95$, and the selection of k will be described in Section 4.3.1.

Evaluation metrics: Following other works of crowd counting, the mean absolute error (MAE) and mean square error (MSE) are used to measure the performance of the network. The definitions of MAE and MSE are shown in formulae below:

$$\text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{C}(I_i) - C(I_i)\right)^2} \tag{12}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{C}(I_i) - C(I_i)\right| \tag{13}$$

where N is the number of testing images, $C(I_i)$ is the GT number of persons in the image I_i, and $\hat{C}(I_i)$ is the predicted number of persons in the image I_i.

The smaller the values of MSE and MAE are, the better the method is.

Table 2 Results of ablation experiments

Configuration	ShanghaiTech part_A		ShanghaiTech part_B	
	MAE	MSE	MAE	MSE
E1:BS + LE	65.4	98.7	9.7	16.0
E2:BD + LE	68.2	108.2	10.2	16.8
E3:BS + BD + LE	64.1	98.8	9.5	15.0
E4:BS + BD + LE + LR(DBSAN)	62.7	96.9	9.3	14.0

Table 3 Results of ResNet50 with different numbers of layers

Configuration	ShanghaiTech part_A		ShanghaiTech part_B	
	MAE	MSE	MAE	MSE
E5:Res22 + LE	99.4	152.2	12.1	20.5
E6:Res40 + LE	75.8	120.0	8.2	14.5
E7:Res49 + LE	76.5	123.1	8.4	14.7

4.3 Experimental results

4.3.1 Ablation experiments: As mentioned in Section 3.3, λ balances the contributions of Euclidean loss and ranking loss. The value of λ will influence the predicted results. Thus, we try different values of λ on the ShanghaiTech part_A dataset; the results are shown in Fig. 8. When $\lambda = 1$, Euclidean loss and ranking loss are equally weighted, the result is best.

We also try different numbers of sub-image k for the ranking loss on the ShanghaiTech part_A dataset; the results are shown in Fig. 9. Although MAE is smallest when $k = 5$, but MAE and MSE are both competitive when $k = 3$. The amount of calculation of the

Table 4 Results of different basic backbone networks

Configuration	ShanghaiTech				UCF_QNRF	
	part_A		part_B			
	MAE	MSE	MAE	MSE	MAE	MSE
E8:Res40 + LE	75.8	120.0	8.2	14.5	116.2	203.0
E9:VGG10 + LE	66.8	99.7	9.7	16.7	112.0	187.0
E10:Res40 + BS + BD + LE + LR	74.8	117.4	8.1	13.3	110.1	181.7
E11:VGG10 + BS + BD + LE + LR	62.7	96.9	9.3	14.0	107.5	176.2

Table 5 Results of different structures on the ShanghaiTech part_A dataset

Configuration	MAE	MSE
E12:VGG10 + B_FPN + LE + LR	82.7	128.5
E13:VGG10 + BS + BD_PSP + LE + LR	66.5	101.8
E14:VGG10 + BS + BD + LE + LR	62.7	96.9

Table 6 Configuration of different structures as the second part of the network

	VGG-backbone (C1–C4 in Table 1)					
B_FPN	BS + BD_PSP				BS + BD	
F5: C-3*3-512	D-3*3-256-2		D-3*3-512-2		D-3*3-256-2	D-3*3-512-2
F6: UP-2*2	D-3*3-128-2		D-3*3-512-2		D-3*3-128-2	D-3*3-512-2
F7: concatenate(F6,C4)	D-3*3-64-2		D-3*3-512-2		D-3*3-64-2	D-3*3-512-2
F8: C-1*1-256	MP-1*1	MP-2*2	MP-3*3	MP-6*6		D-3*3-256-2
F9: UP-2*2	D-3*3-256-2					D-3*3-128-2
F10: concatenate(F9,C3)	D-3*3-128-2					D-3*3-64-2
	D-3*3-64-2					
	C-1*1-1					

network increases as k increases, so $k = 3$ is chosen as our experimental configuration.

To verify the effectiveness of the proposed dual-branch structure and ranking loss function, ablation experiments are done on the ShanghaiTech dataset. We set up four different configurations:

1. Experiment 1 (denoted as E1:BS + LE in Table 2) uses only Branch_S as the second part of the network, and use only Euclidean loss as the loss function.
2. Experiment 2 (denoted as E2:BD + LE in Table 2) uses only Branch_D as the second part of the network, and use only Euclidean loss as the loss function.
3. Experiment 3 (denoted as E3:BS + BD + LE in Table 2) uses the proposed two branches (Branch_S + Branch_D) as the second part of the network, and use only Euclidean loss as the loss function.
4. Experiment 4 (denoted as E4:BS + BD + LE + LR in Table 2) uses the proposed two branches (Branch_S + Branch_D) and use the combination of ranking loss and Euclidean loss as a final loss. Its result is used to evaluate the effectiveness of the proposed ranking loss.

The results of Experiment 1 and Experiment 2 are used as baselines for comparison. The result of Experiment 3 is used to evaluate the effectiveness of the proposed two-branch network structure. The result of Experiment 4 is used to evaluate the effectiveness of the proposed ranking loss. All results of the ablation experiments are shown in Table 2. If we compare the results of Experiment 1 and Experiment 2, one may find that the shallow branch (BS) get better results than the deeper branch (BD), which means lower-level features of heads are more discriminative than higher-level features. This may be because the crowd is dense in the image; the deeper branch cannot extract better feature than a shallow branch. The values of MAE and MSE of Experiment 3 are better than those of Experiment 1 and Experiment 2, which means that the proposed two-branch structure does help to improve the prediction accuracy. The result of Experiment 4 is better than that of Experiment 3, which means the proposed ranking loss is useful for the counting task.

With well-designed architecture and deeper layers, CNNs can perform better in feature extracting. To find out which convolution network is better as a backbone network, we also tried some other structure besides VGG16. Since ResNet is a popular and powerful convolution network, we compare the results of ResNet50 and VGG16.

To find out how many layers of ResNet50 should be used for crowd counting, we try ResNet50 [19] with different numbers of layers as a backbone network. We set up three different configurations:

1. Experiment 5 (denoted as E5:Res22 + LE in Table 3) uses the first 22 layers (corresponding to conv1-conv3_x in ResNet50) of ResNet50 network as the backbone network. Experiment 6 (denoted as E6:Res40 + LE in Table 3) uses the first 40 layers (conv1-conv4_x) of ResNet50 network as the backbone network. Experiment 7 (denoted as E7:Res49 + LE in Table 3) uses the first 49 layers (conv1-conv5_x) of ResNet50 network as the backbone network. All of them are followed by 1×1 convolution to obtain a single-channel predicted density map, and use Euclidean loss as the loss function.

   The results of ResNet50 with different numbers of layers on the ShanghaiTech dataset are shown in Table 3. The values of MAE and MSE on both part_A and part_B of Res40 + LE are better than Res22 + LE and Res49 + LE, so we use ResNet50 with first 40 layers as backbone network when comparing with VGG16 in the following experiments.

We set up four different experiment configurations to compare the results of ResNet50 and VGG16 as different backbone networks:

1. Experiment 8 (denoted as E8:Res40 + LE in Table 4) uses the first 40 layers of ResNet50 network as the backbone network and followed by a 1×1 convolution to obtain a single-channel predicted density map, and use Euclidean loss as the loss function.

2. Experiment 9 (denoted as E9:VGG10 + LE in Table 4) uses the first ten layers of VGG network as the backbone network and followed by a 1×1 convolution to obtain a single-channel predicted density map, and use Euclidean loss as the loss function.
3. Experiment 10 (denoted as E10:Res40 + BS + BD + LE + LR in Table 4) uses the first 40 layers of ResNet50 network as the backbone network, and followed by the proposed dual branches (Branch_S + Branch_D) and use the combination of ranking loss and Euclidean loss as a final loss.
4. Experiment 11 (denoted as E11:VGG10 + BS + BD + LE + LR in Table 4) uses the first ten layers of VGG network as the backbone network, and followed by the proposed dual branches (Branch_S + Branch_D) and use the combination of ranking loss and Euclidean loss as a final loss.

Results of the four experiments are shown in Table 4. The MAE and MSE based on VGG10 are lower on the ShanghaiTech part_A dataset and the UCF_QNRF dataset. The MAE and MSE based on Res40 are lower on ShanghaiTech part_B. To find out the reason, we look into the details of the three datasets. The number of people in ShanghaiTech part_A ranges from 33 to 3139/image, the number of people in ShanghaiTech part_B ranges from 9 to 578/image, and the number of people in UCF_QNRF ranges from 49 to 12,865/image. One may find that the number of people in ShanghaiTech part_B is far less than the number of people in ShanghaiTech part_A and UCF_QNRF, i.e. the crowd scene in ShanghaiTech part_B is relatively sparse. When the crowd is dense, the resolution of each head is low, and deeper network loses more information on small objects. Thus, the deeper network (ResNet50) does not work as good as the simple network (VGG16) in the dense crowd counting task.

In this paper, we propose dual branches, a parallel fully convolutional network (FCN) block as the second part of our network to capture features with different scales. In the experiments, we also tried other architectures as the second part of the network to deal with scale variations. The pyramid pooling module (PPM) in PSPNet [35] and the feature pyramid in feature pyramid network (FPN) [36] are tried for comparison, and three different configurations are set up:

1. Experiment 12 (denoted as E12: VGG10 + B_FPN + LE in Table 5) uses the first ten layers of VGG network as the backbone network and followed by B_FPN as the second part of the network. B_FPN is similar to the feature pyramid. The configuration of B_FPN is shown in Table 6. Moreover, we use the combination of ranking loss and Euclidean loss as a final loss.
2. Experiment 13 (denoted as E13: VGG10 + BS + BD_PSP + LE in Table 5) uses the first ten layers of VGG network as the backbone network and followed by BS + BD_PSP as the second part of the network. The structure of BD_PSP is similar to the PPM. The configuration of BD_PSP + BS is also shown in Table 6. To capture features with different scales, a PPM module, which fused features under four different pyramid scales, is inserted into Block C5_b. Moreover, we also use the combination of ranking loss and Euclidean loss as the final loss.
3. Experiment 14 (denoted as E14:VGG10 + BS + BD + LE in Table 5) is our proposed method, which uses the first ten layers of VGG network as the backbone network, and followed by the proposed two branches (Branch_S + Branch_D) as the second part of the network, and use the combination of ranking loss and Euclidean loss as final loss.

Results of different structures to deal with different scales are shown in Table 5. For the counting task, both MAE and MSE of the proposed dual branches are better than the other two methods.

Table 7 Results on the ShanghaiTech dataset

Methods	part_A		part_B	
	MAE	MSE	MAE	MSE
MCNN [14]	110.2	173.2	26.4	41.3
CNN-based cascaded multi-task learning (CMTL) [37]	101.3	152.4	20.0	31.1
Switch-CNN [18]	90.4	135.0	20.1	30.1
CRSNet [30]	68.2	115	10.6	16.0
SaNet [33]	67.0	104.5	8.4	13.6
perspective crowd counting (PCC) Net [38]	73.5	124.0	11.0	19.0
DBSAN (ours)	62.7	96.977	9.3	14.0

Table 8 Results on the UCF_CC_50 dataset

UCF_CC_50	MAE	MSE
MCNN [14]	377.6	509.1
CMTL [37]	322.8	397.9
Switch-CNN [18]	318.1	439.2
CRSNet [30]	266.1	397.5
SaNet [33]	258.4	334.9
PCC Net [38]	240.0	315.5
DBSAN (ours)	186.8	247.4

Table 9 Results on the UCF_QNRF dataset

UCF_QNRF	MAE	MSE
Idrees *et al.* [31]	315.0	508.0
MCNN [14]	277.0	426.0
CMTL [37]	252.0	514.0
Switch-CNN [18]	228.0	445.0
composition loss (CL)-CNN [31]	132.0	191.0
PCC Net [38]	148.7	247.3
DBSAN (ours)	107.0	176.2

4.3.2 Comparisons with state-of-the-art works: In this section, we compared our approach with some state-of-the-art works [14, 18, 30, 31, 33, 37, 38] on four datasets, ShanghaiTech part_A, ShanghaiTech part_B, UCF_CC_50, and UCF_QNRF.

The experimental results on ShanghaiTech are shown in Table 7, and our approach achieves the best result on part_A. Compared to other methods, our method achieves the lowest MAE and lowest MSE. In terms of MAE, we achieve a decrease of 6.4% comparing with the best result of state-of-the-art works. In terms of MSE, we achieve a decrease of 5.6% comparing with the best result of state-of-the-art works. For part_B, our method achieves a competitive performance in terms of MSE and MAE.

The experimental results on UCF_CC_50 are shown in Table 8. Since this dataset only contains 50 low-resolution crowd images and images contain a wide range of crowd, so the error is the largest compared with other datasets. Compared to other methods, our method achieves the lowest MAE and lowest MSE. The significant improvements to this dataset validate the effectiveness of our methods. In terms of MAE, we achieve a decrease of 27.9% comparing with the best result of other methods. In terms of MSE, we achieve a decrease of 26.1% comparing with the best result of other methods.

The experiment results on UCF_QNRF are shown in Table 9. Our method achieves the best performance in terms of both MAE and MSE. We beat the second-best approach by 18.9% decrease in MAE and 7.8% decrease in MSE.

The density maps of some test images generated by our method on four datasets are shown in Fig. 10. From top to bottom, the original images are from ShanghaiTech part_A, ShanghaiTech part_B, and UCF_CC_50, UCF_QNRF.

5 Conclusion

In this paper, we propose a new CNN structure that performs crowd counting tasks, named DBSAN. The main novelty of the DBSAN is that it consists of two network branches with different depths, which is the key element to solve scale variation problem. We use a basic convolutional network as the backbone and extract

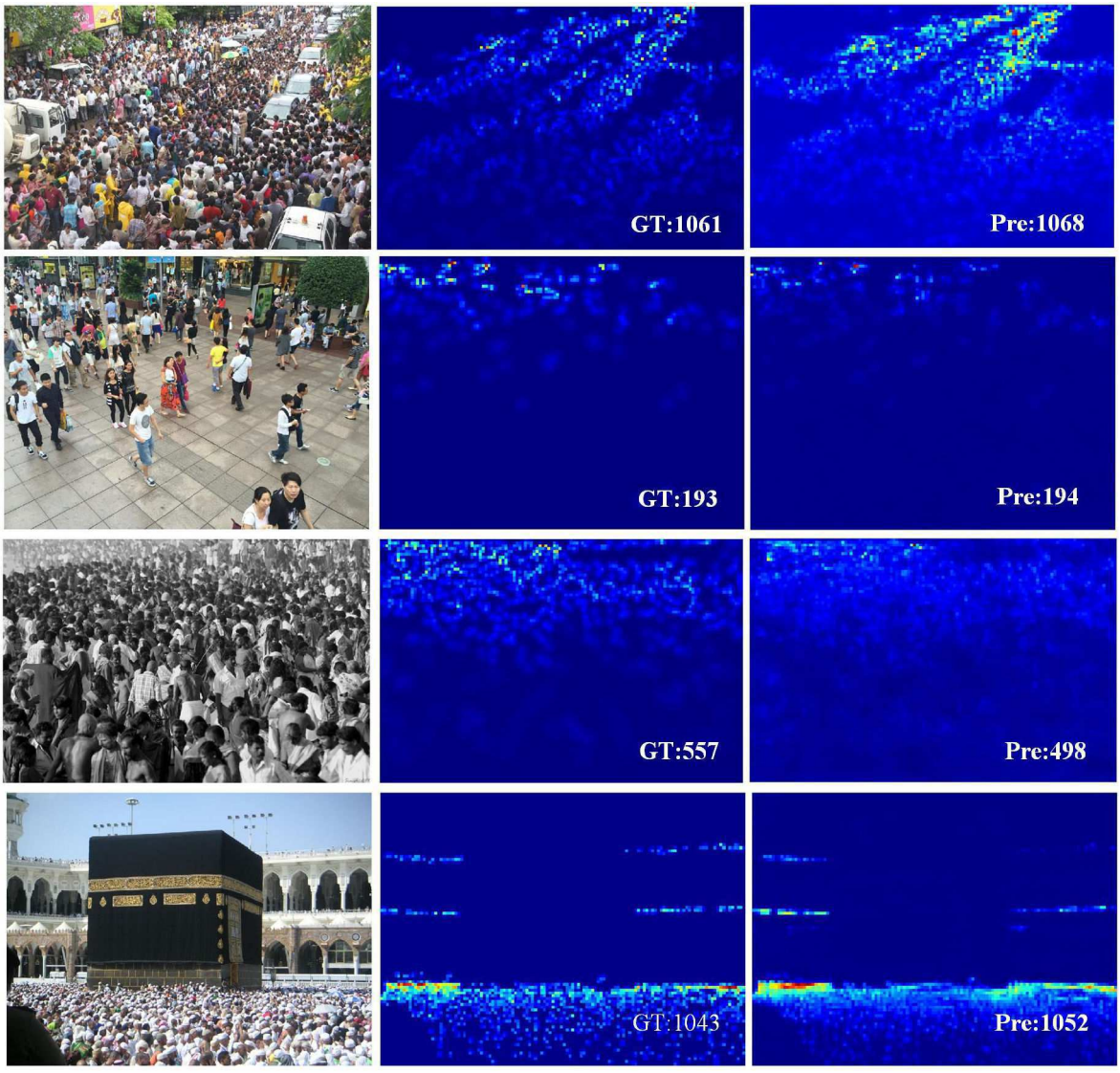

Fig. 10 *Some samples of testing images and density maps on ShanghaiTech part A. From left to right are original images, GT density map, and predicted map by our method. The number of people are marked on the bottom right corners in density map images (GT: ground truth, Pre: prediction)*

features of multi-scale objects by two branches with different depths. On the basis of the observation that an image must contain equal or more persons compared with its sub-image, we propose a novel loss function named ranking loss according to the constraint inside an image. Moreover, we combine the proposed ranking loss with Euclidean loss to get the final loss function for the network. Extensive experiments are conducted on three challenging crowd counting datasets, and the results of experiments show that our method achieves significant improvements over most of the recent state-of-the-art approaches, which demonstrates the effectiveness of our method. In our experiments, we find that crowd images with the complex background will get inaccurate density map. In our next work, we plan to focus on how to reduce the influence of complex background to crowd counting.

6 Acknowledgments

This research was supported by the 111 Project (B12018), the National Natural Science Foundation of China (No. 61972180), and the Natural Science Foundation of Jiangsu Province of China (BK20181341).

7 References

[1] Abdelghany, A., Abdelghany, K., Mahmassani, H., *et al.*: 'Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities', *Eur. J. Oper. Res.*, 2014, **237**, (3), pp. 1105–1118

[2] Chow, W.K., Candy Ng, M.Y.: 'Waiting time in emergency evacuation of crowded public transport terminals', *Saf. Sci.*, 2008, **46**, (5), pp. 844–857

[3] Sime, J.D.: 'Crowd psychology and engineering', *Saf. Sci.*, 1995, **21**, (1), pp. 1–14

[4] Sindagi, V.A., Patel, V.M.: 'A survey of recent advances in CNN-based single image crowd counting and density estimation', *Pattern Recognit. Lett.*, 2018, **107**, pp. 3–16

[5] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., *et al.*: 'Object detection with discriminatively trained part-based models', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, **32**, (9), pp. 1627–1645

[6] Chan, A.B., John Liang, Z.-S., Vasconcelos, N.: 'Privacy-preserving crowd monitoring: counting people without people models or tracking'. 2008 IEEE Conf. Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1–7

[7] Chan, A.B., Vasconcelos, N.: 'Bayesian Poisson regression for crowd counting'. 2009 IEEE 12th Int. Conf. Computer Vision, Kyoto, Japan, September 2009, pp. 545–551

[8] Garcia-Bunster, G., Torres-Torriti, M., Oberli, C.: 'Crowded pedestrian counting at bus stops from perspective transformations of foreground areas', *IET Comput. Vis.*, 2012, **6**, (4), pp. 296–305

[9] Viola, P., Jones, M.J.: 'Robust real-time face detection', *Int. J. Comput. Vis.*, 2004, **57**, (2), pp. 137–154

[10] Dalal, N., Triggs, B.: 'Histograms of oriented gradients for human detection'. 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, vol. **1**, pp. 886–893

[11] Zhang, C., Li, H., Wang, X., *et al.*: 'Cross-scene crowd counting via deep convolutional neural networks'. 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, June 2015, pp. 833–841

[12] Deb, D., Ventura, J.: 'An aggregated multicolumn dilated convolution network for perspective-free counting'. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, June 2018, pp. 308–309

[13] Wang, Z., Xiao, Z., Xie, K., *et al.*: 'In defense of single-column networks for crowd counting'. arXiv: 1808.06133 [cs], August 2018
[14] Zhang, Y., Zhou, D., Chen, S., *et al.*: 'Single-image crowd counting via multi-column convolutional neural network'. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 589–597
[15] Zeng, L., Xu, X., Cai, B., *et al.*: 'Multi-scale convolutional neural networks for crowd counting'. arXiv: 1702.02359 [cs], February 2017
[16] Zhang, L., Shi, M., Chen, Q.: 'Crowd counting via scale-adaptive convolutional neural network'. arXiv:1711.04433 [cs], 2017
[17] Liu, X., van de Weijer, J., Bagdanov, A.D.: 'Leveraging unlabeled data for crowd counting by learning to rank'. arXiv:1803.03095 [cs], 2018
[18] Sam, D.B., Surya, S., Venkatesh Babu, R.: 'Switching convolutional neural network for crowd counting'. arXiv:1708.00199 [cs], 2017
[19] He, K., Zhang, X., Ren, S., *et al.*: 'Deep residual learning for image recognition'. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 770–778
[20] Sun, K., Zhao, Y., Jiang, B., *et al.*: 'High-resolution representations for labeling pixels and regions'. arXiv:1904.04514 [cs], April 2019
[21] Girshick, R.: 'Fast R-CNarXiv N.:1504.08083 [cs]'. arXiv: 1504.08083, April 2015
[22] Liu, W., Anguelov, D., Erhan, D., *et al.*: 'SSD: single shot MultiBox detector'. arXiv:1512.02325 [cs], 9905, 2016
[23] Redmon, J., Divvala, S., Girshick, R., *et al.*: 'You only look once: unified, real-time object detection'. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 779–788
[24] Shen, Z., Xu, Y., Ni, B., *et al.*: 'Crowd counting via adversarial cross-scale consistency pursuit'. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 2018, pp. 5245–5254
[25] Huang, S., Li, X., Zhang, Z., *et al.*: 'Body structure-aware deep crowd counting', *IEEE Trans. Image Process.*, 2018, **27**, (3), pp. 1049–1059
[26] Sam, D.B., Venkatesh Babu, R.: 'Top-down feedback for crowd counting convolutional neural network'. arXiv:1807.08881 [cs], July 2018
[27] Wang, C., Zhang, H., Yang, L., *et al.*: 'Deep people counting in extremely dense crowds'. Proc. 23rd ACM Int. Conf. Multimedia (MM '15), Brisbane, Australia, 2015, pp. 1299–1302
[28] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'ImageNet classification with deep convolutional neural networks'. Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), NY, USA, 2012, pp. 1097–1105
[29] Boominathan, L., Kruthiventi, S.S.S., Venkatesh Babu, R.: 'CrowdNet: a deep convolutional network for dense crowd counting'. arXiv:1608.06197 [cs], 2016
[30] Li, Y., Zhang, X., Chen, D.: 'CSRNet: dilated convolutional neural networks for understanding the highly congested scenes'. arXiv:1802.10062 [cs], 2018
[31] Idrees, H., Tayyab, M., Athrey, K., *et al.*: 'Composition loss for counting, density map estimation and localization in dense crowds'. arXiv:1808.01050 [cs], 2018
[32] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition'. arXiv:1409.1556 [cs], 2014
[33] Cao, X., Wang, Z., Zhao, Y., *et al.*: 'Scale aggregation network for accurate and efficient crowd counting'. Computer Vision (ECCV 2018), Munich, Germany, 2018, vol. **11209**, pp. 757–773
[34] Oñoro-Rubio, D., López-Sastre, R.J.: 'Towards perspective-free object counting with deep learning'. Computer Vision (ECCV 2016), Amsterdam, Netherlands, 2016, vol. **9911**, pp. 615–629
[35] Zhao, H., Shi, J., Qi, X., *et al.*: 'Pyramid scene parsing network'. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 2017, pp. 6230–6239
[36] Lin, T., Dollár, P., Girshick, R., *et al.*: 'Feature pyramid networks for object detection'.arXiv: 1612.03144, April 2017
[37] Sindagi, V.A., Patel, V.M.: 'CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting'. arXiv:1707.09605 [cs], 2017
[38] Gao, J., Wang, Q., Li, X.: 'PCC Net: perspective crowd counting via spatial convolutional network'. arXiv:1905.10085 [cs], 2019