Applied Intelligence (2019) 49:2415–2433
https://doi.org/10.1007/s10489-018-1394-9

Effective use of convolutional neural networks and diverse deep supervision for better crowd counting

Haiying Jiang[1] · Weidong Jin[1]

Published online: 17 January 2019

Abstract

In this paper, we focus on the task of estimating crowd count and high-quality crowd density maps. Among crowd counting methods, crowd density map estimation is especially promising because it preserves spatial information which makes it useful for both counting and localization (detection and tracking). Convolutional neural networks have enabled significant progress in crowd density estimation recently, but there are still open questions regarding suitable architectures. We revisit CNNs design and point out key adaptations, enabling plain a signal column CNNs to obtain high resolution and high-quality density maps on all major dense crowd counting datasets. The regular deep supervision utilizes the general ground truth to guide intermediate predictions. Instead, we build hierarchical supervisory signals with additional multi-scale labels to consider the diversities in deep neural networks. We begin by obtaining multi-scale labels based on different Gaussian kernels. These multi-scale labels can be seen as diverse representations in the supervision and can achieve high performance for better quality crowd density map estimation. Extensive experiments demonstrate that our approach achieves the state-of-the-art performance on the ShanghaiTech, UCF_CC_50 and UCSD datasets.

Keywords Convolutional neural networks (CNNs) · Crowd counting · High-resolution density map · Multi-scale labels · Diversity

1 Introduction

In recent years, the degree of urbanization has substantially enhanced, and the number of urban population has grown exponentially, which has led to an increased number of activities such as sporting events, political rallies, religious gatherings, democratic protests, etc. (see Fig. 1 for various crowd scenes), thereby resulting in massive crowd gathering. In such scenarios, it is essential to analyze crowd behavior for better management, safety and security. Critical to such analysis is crowd count and density.

The history of crowd counting is extremely rich, we now highlight a few representative works that have proven to be of great practical importance. Broadly speaking, one may categorize works into a few groups such as I: monolithic style and part-based pedestrian detection like the appearance and motion feature based pedestrian detection [1], Bayesian model segmentation based crowd detections [2], and the HOG based head detections [3]; methods driven by II: regression-based methods on top of features arrived at through careful manual design, such as foreground area [4–9], texture features [6, 7, 10], histograms of edge orientation [5, 6, 11], or edge count [5, 6, 8]; and III: density based methods that remain reliant on features of the human design, such as MESA [12], random regression forest [13], Count-forest [14], subspace learning [15]. Also, there has been a recent wave of development using Convolutional Neural Networks(CNNs) that emphasize the importance of automatic hierarchical feature learning, including Wang et al. [16], Fu et al. [17] and Zhang et al. [44]. Recent works using multi-column and multi-scale architectures (MCNN [18], Switch-CNN [20], Hydra CNN [21] and CNN-boosting [22]) have demonstrated considerable success in achieving lower count errors. However, there remains large room for improvement in these CNN-based methods, in both performances of networks and run time.

Figure 1 intuitively explains the crowd counting difficulty. In those figures, we see extreme crowding, the size of each

✉ Haiying Jiang
jianghaiying909@gmail.com

Weidong Jin
wdjin@home.swjtu.edu.cn

[1] Southwest Jiaotong University, West Section, High-tech Zone, Chengdu, Sichuan 611756, People's Republic of China

Fig. 1 Sample crowd scene from the ShanghaiTech dateset [18]

person is small. Meanwhile, there is significant variation in the scale of the people who far from the camera appear to be small while the near ones are significantly bigger.

To count such a tight crowd of small objects, one of the most critical elements is the contextual information in an image. Hu and Ramanan [23] showed the importance of contextual information for CNN to recognize small objects. In CNNs, different contextual information is equivalent to requiring different receptive fields. The most common approach to get those different receptive fields is to pool (and subsampling in general) the feature maps throughout the network. However, resolution is lost when subsampling. Resolution is necessary to resolve a tight crowd of small objects. The increased receptive field (and thus recognition capability) comes at the price of losing resolution. The resulting coarse features miss the details of small objects that are difficult to recover even with efforts such as dilated convolutions [24] or encoder-decoder network [25]. Thus, we need a specific method to properly address recognition/resolution tradeoff.

As the promising method, [26] introduced "skip branches" to the fully convolutional neural networks (FCNNs), which add features from the lower convolutional layers to the upsampled layers, to compensate for the loss of spatial information due to the stride in the convolution/pooling layers. Skip networks address the tradeoff between expanding the receptive field and resolution quite explicitly: the information at different resolutions is extracted and combined. The original paper introduces this methodology as "combining what and where." Actually skip-network works quite well in current computer-vision papers [26, 27].

Nevertheless, we highlight that a naive application of skip-network does not always improve performance. The skip model seems to be inflexible and arbitrary in how to combine different resolutions features. First of all, it combines verdicts(e.g. classification maps), instead of a rich set of features. For example, it combines how a layer evaluates that an object is a person using low-level information, with how another layer evaluates whether the same object is a person using higher level information. This means that if objects may not be detected at lower levels, it is useless to combine those verdicts(classification maps) to higher levels. Moreover, the element-wise addition restricts the combination of resolutions to be simply a linear combination. We could certainly imagine that verdicts require a more complex nonlinear combination of high-level and low-level information to be effectively calculated.

We solve this problem by combining a rich set of features, coming from each of the resolutions instead of verdicts. To this end, we propose a novel module, which we call Multilayer Perception(MP) module (Fig. 3). The MP module extracts intermediate features from the network and treated equally, then learns how to nonlinearly combine these features to give the final verdict. This adds flexibility to learn more complex relations between the different resolutions and generalizes the element-wise addition of the skip architecture.

The proposed MP module comprises of a fusion layer which fuses the outputs of multiple branches. Supervision only in the last fusion layer may cause heavy bias towards learning large objects structure, that is, some layers may not be optimized adequately. To alleviate this issue, in this paper, we utilize deep supervision [28] method, namely, both outputs of all branches and their fusion result are supervised. However, using only one general supervision(i.e. the original ground-truth) ignores the network diversities: diverse presentations of hierarchical layers. In addition, the general supervision cannot be well-suited to all branches. Therefore, we propose using diverse deep supervision (DDS) that can adapt to all of branches or hierarchical diversities for crowd counting. DDS mainly utilizes multi-scale labels which vary from coarse level to fine level as deep features become more discriminative. The labels in our work are the density maps which preserve spatial distribution information of crowd. Briefly speaking, we capture the multi-scale density maps and insert them into the network to guide the intermediate layers in a coarse-to-fine side outputs. The multi-scale labels not only make the network more discriminative, but also are a good substitute for perspective maps which are laborious to generate and unavailable for every dataset.

The proposed method uses CNNs to fuse features at various levels for achieving lower count error and incorporate network diversities for better quality density maps. It can be considered as a set of multilayer-perception CNNs to estimate multi-scale density map. Hence, the whole network architecture referred to as Multilayer Perception Counting (MPC).

To summarize, we present several new elements as our contributions: (1) Note that CNNs for accurate crowd counting needs to address the trade-off between recognition and resolution, we analysis the main families of recently proposed single-column CNNs architectures for tackling the aforementioned issue. (2) We design a single-column CNNs with a single small filter size as the front-end for 2D feature extraction, which is easy to train from scratch. Small-sized filters not only preserve the spatial resolution but more effective than large ones and allow us to build a deep network. (3) We combine the feature maps of multiple layers to balance the recognition/resolution tradeoff. Different layers share the same low-level feature representations, which results in fewer parameters, fewer training data required, and faster training. (4) We introduce a diverse deep supervision (DDS) by adding diverse supervision to all side-outputs. It can improve the network generalization and discrimination on crowd scenes with a large variation in crowd density. (5) We conduct experiments on three datasets: ShanghaiTech [18], UCF_CC_50 [55] and UCSD [6]; the results show that our MPC significantly outperforms the state-of-the-art crowd counting methods.

2 Significance and related work

The proposed multilayer-perception counting (MPC) tackles two critical issues: (1) designing and training counting model of the tradeoff between recognition and resolution, the model is to enlarge receptive fields without losing spatial resolutions and brutally expanding network complexity. and (2) model discrimination ability improving with deep diversity supervision, that performs to multiple layers supervision using multi-scale labels guide early estimation results. We discuss below the significance of the proposed MPC algorithm when compared with the existing algorithms along two directions in terms of: (1) Crowd counting and density map estimation; and (2) Tradeoff between recognition/resolution in neural networks

2.1 Crowd counting and density map estimation

The task of crowd counting and density estimation is inherently challenging. After a few decades of research, they have been compounded by myriad of factors that are key and that are likely to play a role in a successful system: (1) carefully designed and/or learned features [5, 6, 29, 39], (2) foreground pixel extraction or background subtraction [30–32], (3) regression model application [9, 12, 33–35] (4) perspective-aware [5, 6, 18, 21, 32, 36] (5) scale-aware [18, 20, 21, 37, 38] (6) whole image predictions (referring to approaches that perform prediction by taking the image contents globally and directly) [18, 19, 38–40], (7) multi-task learning [40–44] and (8) context-aware [19, 39, 45].

Starting with the seminal work of Lempitsky et al. [12], a density-based approach is proposed to learn a linear mapping between local patch features and corresponding object density maps, primarily focuses on three of these aspects: using a large number of manually designed features (property 1), median filtering background subtraction (property 2), and MESA distance regression(property 3). Inspired by the recent deep learning success in computer vision, recent crowd counting tasks are addressed by CNNs. The CNN-based approaches have demonstrated significant improvements over previous hand-crafted feature-based methods, thus, motivating more researchers to explore CNN-based approaches further for related crowd analysis problems. Sam et al. [20–22, 47] using CNNs for patch-based crowd counting contains an alternative common thread that focuses on three aspects: automatic feature learning (property 1), considering perspective-aware (property 4), and multi-scale response fusion (property 5). However, For patch-wise predictions, as in [21], the CNNs normally produce patches of density maps for overlapping image patches. Although the whole density map can be obtained by placing the density patches at their image position and then averaging pixel density values across overlapping patches, but the overlapping prediction and averaging operation results in density maps that are overly smooth(e.g., see Fig. 2c). More importantly, their patch-to-pixel or patch-to-patch strategy results in significantly downgraded training and prediction efficiency.

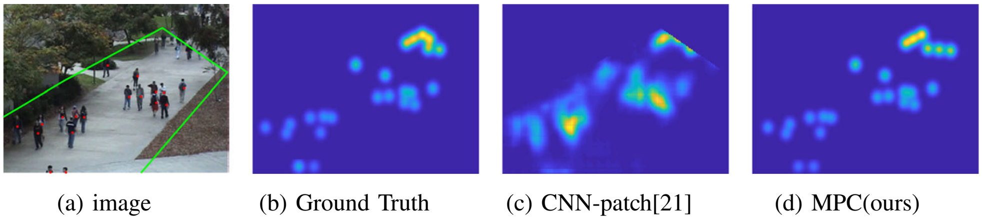
(a) image (b) Ground Truth (c) CNN-patch[21] (d) MPC(ours)

Fig. 2 Comparison of different density map methods. All the density maps are in same color scale (a particular density value corresponds to the same color across images). The green line in the image shows the region of interest (ROI). The red dots are the ground-truth person annotations

By our MPC network architecture, we intend to emphasize that we are producing an easy-trained end-to-end crowd counting system. Zhang et al. [18, 38, 43, 45] also perform a whole-image based inference, which focuses on four aspects: automatic feature learning (property 1)multi-scale response fusion(property 5) making whole image predictions (property 6), exploiting multi-task learning(property 7). Crowd counting via regression suffers from the drastic changes in perspective and scale, which commonly exist in crowd images (see Fig. 1). Marsden et al. [38] attempted to address the scale issue by performing a multi-scale averaging during the prediction phase. While being simple and effective, it results in an inefficient inference stage. Zhang et al. [18] uses multi-column of convolution layers with the same depth but with different filter sizes(small, medium and large), which are combined in the end to adapt to the large variations of perspective and scale. Despite [18] has achieved high performance in this field, we observe two disadvantages of it: (1) It is hard to train according to the described training method. (2) It used 9 × 9 filters in order to get a larger receptive field (with fixed depth), which is not as effective as smaller filters [46]. Instead, we use single-column CNNs with a small filter as the backbone, which is easy to train. Meanwhile, small-sized filters preserve the spatial resolution and are more effective.

In an entirely different approach, Sindagi et al. [43, 45] explored multi-task learning to boost individual task performance. Vishwanath et al. [43] used a cascaded CNN architecture to simultaneously learn to classify the crowd count into various density levels and estimate density map. However, using a density-level classifier makes the design more complicated and spend a large portion of network parameters. To generate high-quality density maps, Vishwanath et al. [45] adopts multi-column convolutional neural networks (MCNN) as a part of their network and further concatenates its feature maps with local and global context features from classification networks. While their approach is successful in generating high-quality density maps, they run much slower since it uses both a bloated network structure and sliding window fashion per-pixel prediction. Compare to our MPC, we use a single backbone network with a single filter size. Density map estimation is performed at a concatenated layer which merges the feature maps from multiple scales of the network; it can therefore easily adapt to scale variations and address the recognition/resolution tradeoff. In addition, multi-scale layers share the same low-level parameters and feature representations, which results in fewer parameters and faster prediction.

Works similar to ours are SaCNN [51] and FCNN-skip [40]. The most obvious differences between our network with SaCNN and FCNN-skip are in two part. First, SaCNN and FCNN-skip only consider the last few convolution layer feature to combine, in which lots of local information to crowd density estimation is missed. In contrast to it, our MPC uses richer features from all the convolution layers, thus it can capture more accurately crowd location information. Second, our architecture comprises a single-stream deep network with additional diverse deep supervision. Compared with traditional deep supervision, diverse deep supervision can adapt to the hierarchical diversities with minimal manual effort. This architecture can improve both discrimination and generalization for high-quality density maps generation.

2.2 Tradeoff between recognition/resolution in neural networks

As pointed out in [45], context information provide significant improvements in the crowd counting. Even humans cannot recognize a small person in a surveillance imagery patch without context information such as caps, bags or other persons. Also, a higher spatial resolution is crucial. In coarse resolution, small crowd can be over-estimated, or under-estimated. Thus, we should pay attention to both context(recognition capability) and resolution. But, as we know, it seems contradictory to try to increase recognition along with resolution in general signal-column CNNs. Fortunately, there's been a lot of work on how to deal with the balance between recognition and resolution. In the following, we analyze recently the main families of addressing the recognition/resolution tradeoff networks that have been used in the past two years.

Dilation networks Dilated convolutions are basically convolutional filters with gaps between the filter elements. By increasing this gap, the kernel weights are placed far away at given intervals (i.e., more sparse), and the kernel size accordingly increases. Dilations have been used as an alternative to upsampling for generating full-resolution outputs [26, 52]; or as a means to increase the receptive field [53], by enlarging the area covered by a convolution kernel without increasing the number of trainable parameters.

While dilation networks have been reported to exhibit certain advantages, they are computationally demanding. Application of dilated convolution causes the problem: the receptive field is indeed increased, but spatial consistency between neighboring units becomes weak and local structure cannot be extracted in a higher layer.

Deconvolution networks (Unpooling) The deconvolution scheme uses a series of layers to learn to interpolate and upsample the output. This is usually formulated as an encoder/decoder architecture, where a base network is first designed (the encoder) and then a reflected version of itself is attached to it (the decoder, with corresponding deconvolution and unpooling layers). However, the depth of deconvolution networks is significantly larger, roughly twice the one of the associated FCN. This often implies a slower and more difficult optimization, due to the increase in trainable parameters introduced by deconvolutional layers. This model does address the recognition/resolution tradeoff, but only in the case where max pooling is used for subsampling. To recover the lost resolution, it is required to transmit the indices of the maximal activation of the max pooling layers to the corresponding decoder unpooling layers.

Skip networks Skip networks as the name suggests skips some layer in the neural network and obtain intermediate features at different resolutions (not just the last one), the final output map is built by combining multiple feature responses. Skip networks add the link to incorporate the feature responses from different resolutions of the primary network stream, and these responses are then combined in a share output layer. The skip network architecture provides an efficient solution to address the recognition/resolution tradeoff. However, the model is arbitrary and inflexible in how the features are combined.

Multilayer-perception counting networks We list these variants to help clarify the distinction between existing approaches and our proposed multilayer-perception networks approach, illustrated in Fig. 3. There is often significant redundancy in existing approaches, in terms of both representation and computational complexity. Our proposed multilayer-perception counting network take multiple intermediate features at different resolutions and combine them seems to be a sensible approach to specifically address the recognition/resolution tradeoff. In such a scheme, MPC can expand the receptive field without losing resolution, meanwhile combining all of intermediate features constitutes indeed an efficient use of resources.

3 Our approach

3.1 Network architecture

In this paper, we design a MPC network for counting crowd and generating high-quality density maps. Unlike the latest works such as [20, 45] which use the deep CNN for ancillary, we focus on designing CNN-based density map generator. Figure 3 shows a schematic of the proposed MPC model: front-end module, diverse deep supervision(DDS) module and multilayer perception(MP)module. The role of each module is different. The front-end module is designed to extract features that cover large context, and thus the size of the receptive field is gradually increased as the network deepen. The lateral MP module is dedicated to

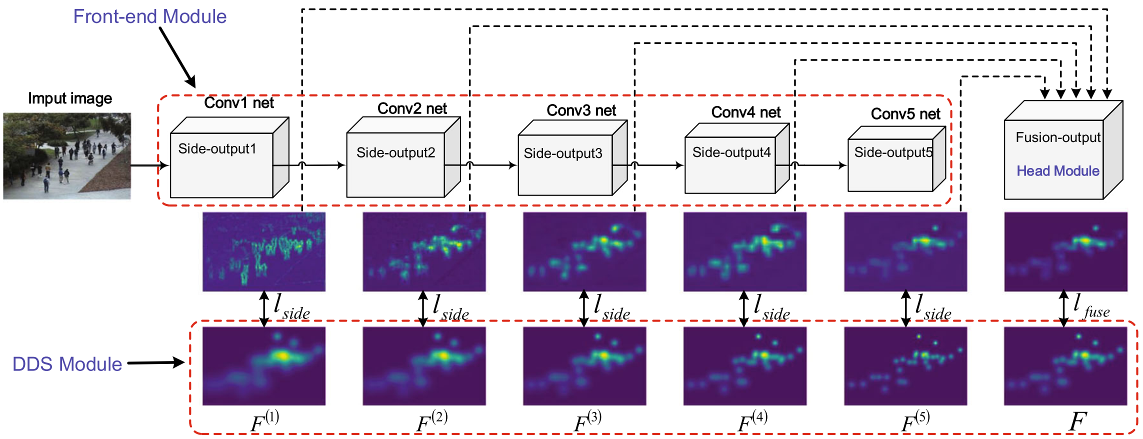

Fig. 3 Overview of the proposed network architecture

aggregating hierarchical features scattered by the front-end module. Thus, the MP module is constructed using a set of deconvolutional and fractionally-strided convolutional layers. The set of fractionally-strided convolutional layers help us to restore details in the output density maps. Finally, The presence of DDS module in MPC is designed to improve the generalization and discrimination of the model. Thus, the DDS module has discriminating layer connected to the deconvolutional layer in each stage.

Front-end Front-end module is our backbone. We need the architecture of Front-end (1) to be deep, so as to efficiently generate perceptually multi-level features; (2) to have multiple stages with different strides, so as to generate meaningful side outputs with different scales; and(3)to be easy-trained. Recently, VGG16 nets [48] have gained popularity for computer vision tasks [26, 49, 50], with great depth (16 convolutional layers), great density (stride-1 convolutional kernels), and multiple stages (five 2-stride downsampling layers). Recent work [46] also demonstrates that carving the first 13 layers from VGG16 is useful to predict the density map for a given crowd image. We therefore adopt the VGG16 architecture but make the following modifications: (a) we remove the classification part of VGG16 (fully-connected layers) and build the proposed MPC with convolutional layers in VGG16. (b)we cut the 5th pooling layer of VGG16. If we add 5th pooling layer, the output size of this front-end network is 1/32 of the original input size. That would be harmful for crowd localization and generating high-quality density maps. Our final Front-end module has 5 stages, with strides 1, 2, 4, 8 and 16, respectively, and with different receptive field sizes, all nested in the VGG16. It is clear that the useful information captured by each *conv* layer becomes coarser with its receptive field size increasing. However, simple extension of receptive field fails to generate clear density map because of spatially abstracted coarse features. The problem is solved in next section of the network.

Multilayer perception(MP) The role of MP module is to handle problem of front-end module. Specifically, aggressive application of pooling causes: the high-resolution features have a small receptive field, while the low-resolution ones have a wider receptive field. If we combine the different resolution features to adapt the network to the changes in receptive field size, spatial resolution in crowd images can be maintained. In such a scheme, we extract a subset of intermediate features from the front-end module, then concatenate those to create the pool of features. From the pool of features, a neural network predicts the final density map. In experiments section, the MP module is shown to be effective especially for densely crowd scene.

Diverse deep supervision(DDS) Lee et al. [28] firstly proposed to train deep neural networks with hidden-layer supervision. Although they imposed additional supervision to intermediate layers to improve the directness and transparency of learning network, their general supervision fails to present hierarchical diversities. Instead, our main aim is to explicitly introduce diversities associated with different intermediate layers. To this end, we propose the diverse deep supervision, which acts as multiple specific discriminant layers that produces a companion local output map for early layers. The main difference of the diverse deep supervision with [28] is: Notably, Lee et al. utilizes the original ground-truth as supervisory signal to guide the whole network. Instead, we use multi-scale ground-truth labels which vary from coarse-to-fine as deep features become more discriminative to instead of the original ground-truth. The multi-scale ground-truth make the network more discriminative and perform accurate count estimation as well as present high quality density maps.

3.2 Multi-scale labels generation

In the training images, $I_i, i = 1, \cdots N$, all objects presented in the image must be annotated with one point in the center. The true density ground-truth for each pixel $p \in I_i$ is defined as a sum of Gaussian kernels centered on the point annotations :

$$F_i = \sum_{P \in P_i} N(p; P, \sigma^2 I), \forall p \in I_i \tag{1}$$

where p is a pixel location of image I_i, $N(p; P, \sigma^2 I)$ is a $2D$ Gaussian distribution centered at P, and P_i is the set of ground-truth object locations in image I_i. The parameter σ is the standard deviation of Gaussian filter. Depending on the dataset, we use different methods to determine parameter σ for generating ground truth density. Due to perspective distortion and cross-scene scenarios, the images usually contain heads of very different sizes. For the sparse scenes, the parameter σ is the average head size. For the highly congested scenes, we describe the process to estimate the parameter σ as:

$$\begin{aligned} \sigma_i &= \arg_{P \in P_i} \beta \overline{d_i(P)} \\ \sigma &= \frac{1}{N} \sum_{i=1}^{N} (\sigma_i) \end{aligned} \tag{2}$$

For each targeted object in the ground truth P_i, we use $\overline{d_i(P)}$ to indicate the average distance of k nearest neighbors. $\beta\overline{d_i(P)}$ is geometry-adaptive kernels [18] which vary the spread parameter of the Gaussian depending on the local crowd density. Hence σ_i is the average spread of Gaussian calculated from all objects in an image. For N training images, σ in this case is given by averaging the σ_i. In experiment, we refer to the configuration in [18], where $\beta = 0.3$ and $k = 3$.

In this paper, we developed a crowd counting method influenced by diverse deep supervision(DDS) to guide the CNNs model. Our diverse deep supervision consists of multi-scale labels. These multi-scale labels are used to adapt to the diversities of intermediate layers which can present different scale abstracts of the input image. Multi-scale labels can be obtained by convolving the annotation map with Gaussian kernels of different standard deviation σ^j (j is the number of side-outputs). The method of calculating the parameter σ^j is the same as the previous method of calculating the parameter σ. For datasets with dense crowd, we first compute $\overline{d_i(P)}$ in the dataset and sort them in ascending order, then divide them into j parts. Finally, we average the jth part of $d_i(P)$ to calculate the value σ^j. Similarly, this approach is equally applicable to datasets that have sparse crowds. The corresponding multi-scale labels are denoted as $\{F^j\}_{j=1}^J$, where $J = 5$ in this network. Five different side-output predictions are separately supervised with the corresponding F^j, and the fusion-output prediction is still supervised with the original ground-truth (see Fig. 3).

3.3 Network Loss

For MPC training, we have 5 side-output layers and a fusion layer, each side-output layer is also associated with a regressor. We consider the squared Euclidean distance loss to measure the distance between the estimated density map and the ground truth [18, 20]:

$$\ell_{density}(d, \hat{d}) = \frac{1}{N}\sum_{i=1}^{N}\left(d_i - \hat{d}_i\right)^2 \tag{3}$$

d_i and $\hat{d}_i$ denote the ground truth density value and the predicted density value respectively, N is the number of training images. The Euclidean distance is computed at pixels and summed over them.

Putting side-output layer loss and fusion layer loss together, we minimize the following objective function via standard (back-propagation) stochastic gradient descent:

$$\ell = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{|J|}\ell_{side}(d_i^j, \hat{d}_i) + \ell_{fuse}(d_i, \hat{d}_i^{fuse})\right) \tag{4}$$

where d_i^j is the activation value from stage j while $\hat{d}_i^{fuse}$ is from fusion layer. J is the number of stages (equals to 5 here).

We introduce another loss functions to jointly optimize the model: count loss. The count loss helps to reduce the variance of the prediction errors. The count loss function is

$$\ell_{count} = \frac{1}{N}\sum_{i=1}^{N}\left(\Sigma d_i - \Sigma\hat{d}_i\right)^2 \tag{5}$$

The crowd counting loss concentrates the learning on those samples with relatively large prediction errors.

The training and evaluation was performed on NVIDIA GeForce GTX 1080 GPU using Caffe framework [54]. We directly train our MPC network from scratch by randomly initializing the network parameters. Stochastic gradient descent (SGD) minibatch samples 10 images randomly in each iteration and applies fixed learning rate at $1e-6$ during training. The momentum and weight decay are set to 0.9 and 0.0002 respectively. We first regress the model on the density map (4); once it converges, we add (5) in the objective loss to jointly train a multi-task network for a few more epochs.

4 Experiments

In this section, we discuss our detailed implementation and report the performance of our proposed algorithm. The data preparation and evaluation metrics are introduced, and then we evaluate our method on three benchmarks of crowd counting datasets. The first one, ShanghaiTech dataset [18], which consists of two parts as Part A and Part B. Part A is firstly used to establish an ablation study for validating the effects of different modules, and then the whole dataset (Part A and Part B) is conducted to compare with the previous state-of-the-art methods. The other two, UCF_CC_50 dataset [55] and UCSD Pedestrian dataset [6]are used to benchmark our method with the previously proposed crowd counting methods. The statistics of the datasets are summarized in Table 1, and some qualitative results can be found in the supplementary.

4.1 Data preparation

To create the training dataset, we divide the input image into four quarters blocks without overlapping. After that, we also crop patches of size $1/4^{th}$ the size of the original image from 60 random locations. The cropping leads the training set that is a factor of 64 larger than the original dataset. Note that the cropping is used only as a data augmentation technique and the resulting patches are of arbitrary sizes.

Among these datasets, the UCF_CC_50 and ShanghaiTech Part A datasets are congested scenes, UCSD and

Table 1 Statistics of the four tested datasets

Dataset		N_f	Res.	Range	T_P
UCSD		2000	238 × 158	11-45	49885
UCF_CC_50		50	varies	96-4633	63974
ShanghaiTech	Part A	482	varies	33-3139	241,677
	Part B	716	768 × 1024	9-578	88488

N_f is the number of annotated images or frames; Res. is the image/frame resolution; Range is the range of number of objects inside the ROI of a frame; T_p is the total number of labeled objects

ShanghaiTech Part B datasets are sparse scenes. According to the description in Section 3.2, we use different method to calculate the parameter σ of Gaussian kernel depending on the dataset. The setups for different datasets are shown in Table 2.

4.2 Evaluation metric

In our experiments, the following three metrics (7), (8) and (9) to measure the prediction accuracy of the model: the mean absolute error (MAE), Mean Squared Error(MSE), and standard deviation of absolute error (Std_AE) [56], which is used to quantify the amount of variation or dispersion of a set of estimation errors values.

$$AE = |c_i - \hat{c}_i| \tag{6}$$

$$MAE = \frac{\sum_{i=1}^{N} AE(i)}{N} \tag{7}$$

$$MSE = \sqrt{\frac{\sum_{i=1}^{N} AE(i)^2}{N}} \tag{8}$$

$$Std\_AE = \sqrt{\frac{\sum_{i=1}^{N}(AE(i) - MAE)^2}{N}} \tag{9}$$

where AE in (6) is the absolute error, and $\hat{c}_i$ is the predicted value obtained from the model, while c_i is the actual value measured, N is the number of samples used for model training, validating or testing. Roughly speaking, MAE indicates the accuracy of the estimates, Std_AE indicates the uncertainty of the estimates, and MSE indicates the robustness of the estimates.

Besides, we also use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity in Image(SSIM)[57] to measure the quality of density maps with respect to the ground truth density map on ShanghaiTech Part A dataset.

4.3 Ablation Study on ShanghaiTech Part A

Here, we discuss the design of our model with ablation studies on ShanghaiTech dataset Part A [18]. In this dataset, people in the images present large variations in density, scale and appearance, so it is difficult to estimate the count with high degree of accuracy. Hence, it was chosen for the detailed analysis of performance of the proposed architecture. In the experiments, we use the train-test splits provided by the authors [18].

Our experiments comprise two axes: MP module(with or without), diverse deep supervision(DDS, with or without). (1) The architecture of the model only uses front-end module(without DDS and MP modules). The output feature maps of front-end are fed into 1 × 1*conv* layer whose output is used to estimate the density map. (2) Front-end with diverse deep supervision(Front-end+DDS). In addition to the front-end configuration, a Euclidean loss layer is also connected to the *deconv* layer in each stage and the network is trained using $\ell_{density}$ loss. (3) Front-end with only MP module(Front-end+MP). The output of Front-end is concatenated with all side-outputs and the network are trained to estimate the density maps by minimizing ℓ_{fuse} loss. (4) Front-end with MP module and diverse deep supervision(front-end+MP+DDS)train end-to-end. Detailed architectures of main models are shown in Table 3. In all

Table 2 The setups of generating the multi-scale labels and ground truth for different datasets

Dataset		Multi-scale labels					Ground truth
		σ^1	σ^2	σ^3	σ^4	σ^5	σ
UCSD		2	3	4	6	8	4
UCF_CC_50		2	3	5	7	9	5
ShanghaiTech	Part A	2	4	6	8	9	6
	Part B	5	7	10	12	15	10

Table 3 Detailed architectures of the networks

Detailed architectures of the networks			
Front-end	MP	DDS	
stage 1		conv-n1-k1	loss
stage 2	deconv-n128-k4	conv-n1-k1	loss
stage 3	deconv-n256-k8	conv-n1-k1	loss
stage 4	deconv-n512-k16	conv-n1-k1	loss
stage 5	deconv-n512-k32	conv-n1-k1	loss
Four different configurations			
Front-end	Front-end+MP	Front-end+DDS	Front+MP+DDS
	concat		concat
	conv-n512-k3		conv-n512-k3
	conv-n256-k3		conv-n256-k3
conv-n1-k1	conv-n1-k1	conv-n1-k1	conv-n1-k1
loss	loss	loss	loss

In the table, the top half of the table is the detailed structure of the main model, and the lower half of the table is the four different configurations of the ablation study. Front-end module is fine-tuned from VGG-16 net where all convolutional layers are divided into five stages, stage(c) represents the c stage. In this table, (de)conv-n(a)-k(b) represents a convolutional layer with $b \times b$ kernel, output number of feature maps of a. loss represents Euclidean distance loss

of the models, convolutional layers except the last one are followed by ReLU activations. All networks take the whole image as input and predict the density map with the same resolution.

Results discussion As we see in Table 4, Front-end+MP+DDS(MPC) achieves the lowest count errors and the highest quality metrics of the estimated density images, and with MP module performs better than without MP modules. In terms of using DDS module, Front-end+DDS performs better than Front-end. Figure 4 shows estimated density maps from various configurations along with [18] on sample input images.

It's natural to think that the single front-end module has very weak performance(row two of Table 4). In a typical pipeline of a counting by regression model, if accurate results are to be achieved, the input features typically require geometric correction to handle the differences in pedestrian size due to the camera perspective and pedestrian velocity. This phenomenon has been described in several works, reporting state-of-the-art results (e.g. [44, 45, 55]). The role of front-end module is to aggregate large context. The features learned by single Front-end are not adaptive to (hence the overall network is robust to) large variation in people/head size due to perspective effect or across different image resolutions. As a consequence, erroneous results are expected by using a *single* front-end module.

Table 4 Estimation errors for different configurations of the proposed network on Shanghai Tech Part A dataset [18]

	Part A			
Architecture	MAE	MSE	PSNR	SSIM
MCNN [18]	110.2	173.2	20.91	0.52
Front-end	135.2	172.7	20.45	0.49
Front-end+DDS	112.6	152.3	20.90	0.54
Front-end+MP	98.9	135.4	20.96	0.58
Front-end+MP+DDS(MPC)	**83.9**	**128.6**	**21.24**	**0.62**

Bold entries represent the best results obtained from different methods

The result on the benchmark dataset (row five of Table 4) differs not only in MAE and MSE but also displays rapidly rise in PSNR and SSIM comparing with Front-end. Here it shows that diverse deep supervision is effective to obtain desired density maps. The key characteristic of our proposed DDS is that each diverse supervision layer is supposed to play a role as a supervisor responsible for guiding network. With DDS, direct control and guidance across multiple scales, this network will not be biased towards learning large objects structure.

As illustrated in Table 4, network structure of Front-end+MP is a little less effective than MPC but clearly much better than Front-end+DDS. The crowd images are more crowded in Part A dataset, pedestrian heads are therefore quite small; whilst the feature maps from the deep layers tend to fire on big heads. Thus combining multi-scale outputs(+MP) doesn't result in a lowest level in MAE and MSE on Part A. But still, if we compare both the Front-end+DDS

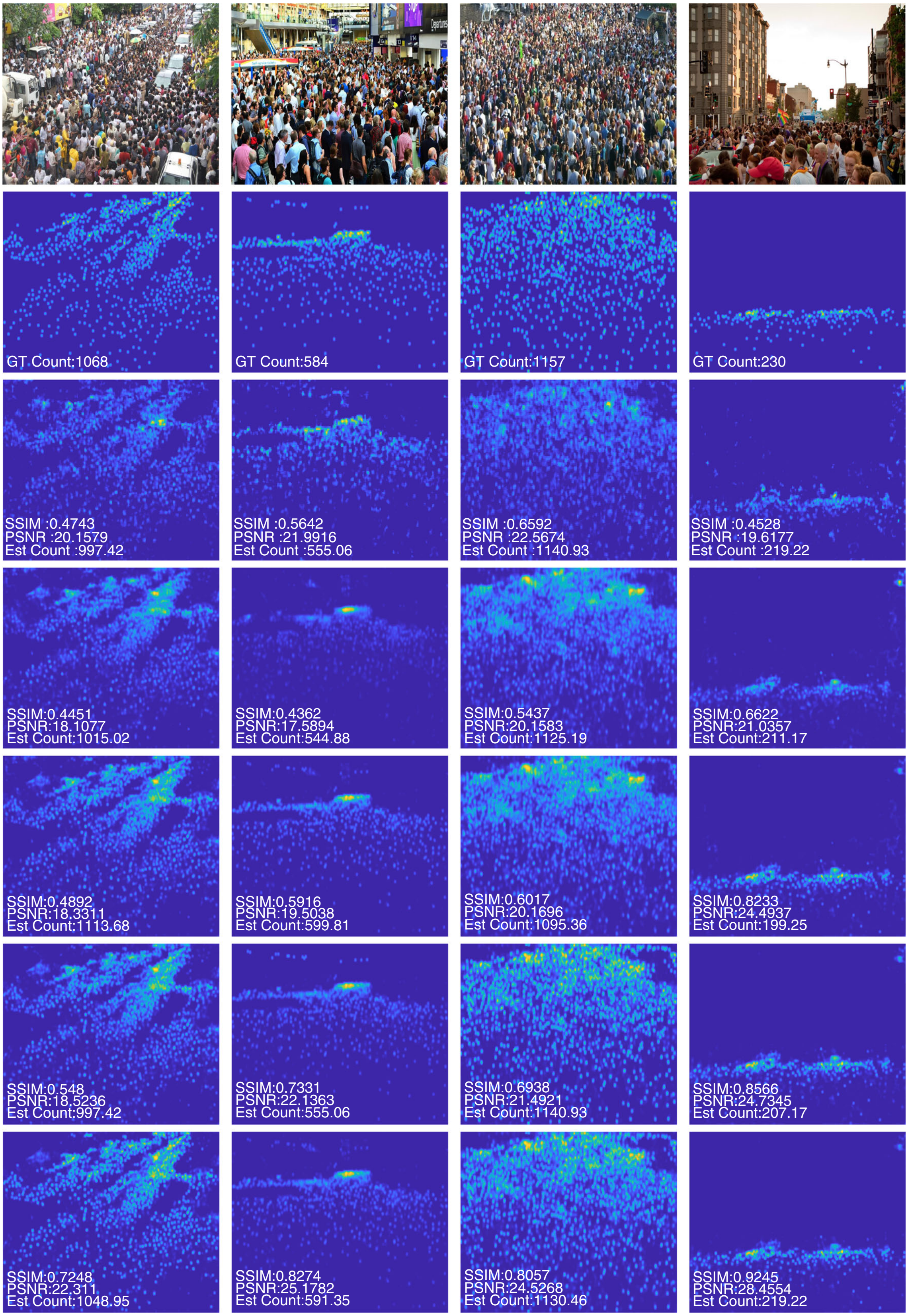

Fig. 4 Comparison of results from different configurations of the proposed network along with MCNN [18]. Top Row: Sample input images from the ShanghaiTech dataset [18]. Second Row: Ground truth. Third Row: MCNN [18]. Fourth Row: Front-end. Fifth Row: Front-end+DDS. Sixth Row: Front-end+MP. Bottom Row: Front-end+MP+DDS(MPC)

and Front-end, Front-end +MP gets a better performance as crowd counter. The results validate our argument that the combination of features from different levels from a network is helpful in better recovering lost spatial information.

Figure 4 shows estimated density maps from various configurations along with [18] on sample input images. It can be observed that the density maps generated using Zhang et al. [18] and Front-end (which regress on low-resolution maps) suffer from loss of details. The use of diverse deep supervision results in better estimation quality. Additionally, the use of multiscale features and minimization over a weighted-fusion of l_{side} and l_{fuse} further improves the quality and reduces the estimation error.

Effect of MP module To further analyze the effect of MP module, we build upon configuration in Fig. 5. We train 5 independent networks induces from the five convolutional blocks of VGG16 nets. Those networks correspond to different depths, where the number of convolutional layers in those networks are 2, 4, 7, 10, 13, respectively. During test, the final prediction is obtained by averaging the outputs of these individual networks. These independent networks achieve (MAE=108.4, MSE=171.3) where-as MPC achieves (MAE=83.9, MSE=128.6) under the same experimental conditions. Moreover, MPC is the nested multi-scale architecture where multi-scale layers share the same low-level parameters and feature representations, which results in fewer parameters, fewer training data required, and faster training.

Diverse deep supervision analysis To explicitly validate the diverse deep supervision availability, we also employ another variant which has the same architecture as our MPC apart from imposing the general supervision(i.e. original ground-truth) to five side-output predictions, that is $\sigma^j = \sigma, j = 1, ...5$. The variant achieves (MAE= 91.7, MSE=135.2). Relatively, the MPC reduces the error count to (MAE=83.9, MSE=128.6), which get 9% lower MAE than the variant. This provides evidence of the benefits of diverse deep supervision.

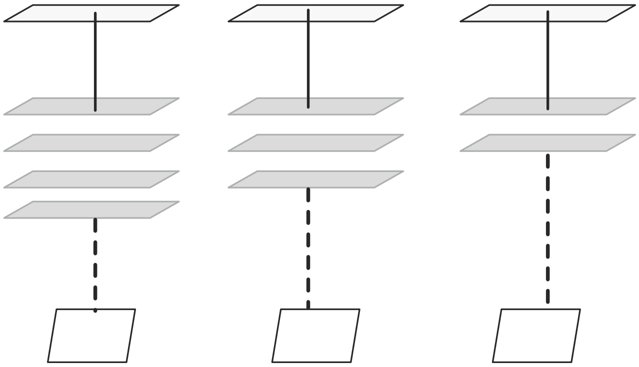

Fig. 5 Separate training of different networks

4.4 Evaluations and comparisons

In this section, we position the proposed method with respect to other results reported in the literature, complete the evaluation on three challenging datasets. Some qualitative results are shown in supplementary material.

ShanghaiTech The ShanghaiTech dataset consists of 1198 annotated images(Part A 482 images and Part B 716 images) with a total of 330,165 people with head center annotations. Following [18], we use 300 images for training in Part A; 400 images for training in Part B. Rest of the images are used as test set. The setups of generating the multi-scale labels and ground-truth density maps on Part A and Part B can be found in Table 2.

On the ShanghaiTech dataset, our method is evaluated and compared to other six recent works: Zhang et al.[44], Marsden et al. [38], MCNN [18], Cascaded-MTL [43], Switching-CNN [20] and CP-CNN [45]. The results are shown in Table 5. It can be observed that our method achieves the best MAE 16.2 and MSE 25.8 on Part B, and a comparable result (second best) on Part A, with only CP-CNN [45] performing better. Note that CP-CNN uses a pre-trained VGG16 network as a density level classifier and sliding-window for per-pixel prediction, which is very slow for images of high resolution, taking 66 seconds to process one image (0.015 fps). In contrast, Our MPC runs at 33.5 fps on 1024 × 768 images (see Table 9). We also report the uncertainty of our model on ShanghaiTech dataset in Table 8.

UCF_CC_50 The UCF_CC_50 dataset [55] is an extremely challenging dataset, contains 50 very different resolutions

Table 5 Comparison of MPC with other state-of-the-art on ShanghaiTech dataset

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Zhang et al. [44]	181.8	277.7	32.0	49.8
Marsden et al. [38]	126.5	173.5	23.8	33.1
MCNN [18]	110.2	173.2	26.4	41.3
Cascaded-MTL [43]	101.3	152.4	20.0	31.1
Switching-CNN [20]	90.4	135.0	21.6	33.4
CP-CNN [45]	**73.6**	**106.4**	20.1	30.1
MPC(ours)	83.9	128.6	**15.8**	**25.2**

Bold entries represent the best results obtained from different methods and aspect ratios images randomly crawled from the Internet. There is a large variation in the number of people varying from 96 to 4,633. The ground-truth density map is generated using $\sigma = 5$. The multi-scale ground-truth density maps are generated using $\sigma \in \{2, 3, 5, 7, 9\}$. Following [44, 55], the evaluation is based on 5-fold cross-validation.

In Table 6, we compared the performance of our models with ten recent approaches on UCF_CC_50 dataset: Idrees et al. [55],Zhang et al. [44], MCNN [18], Hydra 2s [21], CNN Boosting [22], Marsden et al. [38], Cascaded-MTL [43], Switching-CNN [20], FCNN-skip [40]and CP-CNN [45]. Among those methods, CP-CNN [45], taking global and local context information into consideration, performs the best. CP-CNN consists of four modules, each of them is composed with one or multiple columns CNNs. Although such a model has high density estimation accuracy, the training process is quite complicated and computationally demanding. Our MPC requires less computation compared to CP-CNN and also achieves rather high accuracy, its MAE is 313.6. The MAE on this dataset (see Table 6) is relatively high as the dataset has very few training examples and wide variations in the background and crowd density. This also limits the ability of our MPC to learn the diversity of space of crowd scene and causes the difference between the estimate and its ground truth to be spread out over a wider range of mean absolute error (see Table 8).

UCSD The UCSD dataset [6] is composed of 2000 low resolution (238 × 158) surveillance video frames and 49,885 annotated pedestrians on pedestrian walkways at UCSD. The scenes are characterized by sparse crowd with the number of people ranging from 11 to 46 per frame. Training and test date splits used by traditional setting [6]. Of the all frames, frames 601-1400 are used for model training and the remaining 1200 frames are for testing. The ground-truth density map is generated using $\sigma = 4$. The multi-scale ground-truth density maps are generated using $\sigma \in \{2, 3, 4, 6, 8\}$.

Table 6 Comparison of MPC with other state-of-the-art on UCF_CC_50 dataset

Method	MAE	MSE
Idrees et al. [55]	419.5	541.6
Zhang et al. [44]	467.0	498.5
MCNN [18]	377.6	509.1
Hydra 2s [21]	333.7	425.2
CNN Boosting [22]	364.4	341.4
Marsden et al. [38]	338.6	424.5
Cascaded-MTL [43]	322.8	397.9
Switching-CNN [20]	318.1	439.2
FCNN-skip [40]	431.6	379.6
CP-CNN [45]	**295.8**	**320.9**
MPC(ours)	313.6	342.8

Bold entries represent the best results obtained from different methods

Table 7 Comparison of MPC with other state-of-the-art on UCSD dataset when using the whole training set

Method	MAE	MSE
Zhang et al. [44]	1.60	3.31
MCNN [18]	**1.07**	**1.35**
CNN Boosting [22]	1.10	-
Switching-CNN [20]	1.62	2.10
FCNN-skip [40]	1.22	2.25
MPC(ours)	1.12	1.76

Bold entries represent the best results obtained from different methods

The results on UCSD dataset are shown in Table 7. The proposed method is evaluated against five recent state-of-the-art approaches: Zhang et al. [44], MCNN [18], CNN Boosting [22], Switching-CNN [20] and FCNN-skip [40]. Zhang et al. [18] proposed a multi-column convolutional network (MCNN) to adapt to the large variations of perspective and scale and have shown robust performance(lowest MAE 1.07), but the designs they used multi-column architectures also introduce disadvantage of redundant structure. We notice that using one single column is able to retain over 70% accuracy of the multi-column model in [20], where they use three CNN regressors the same as in MCNN. Meanwhile, they use lager filters (9 × 9) in order to get a larger receptive field (with fixed depth), which is not as effective as smaller filters. Finally, such multi-column CNNs architecture requires more time to train. Although MCNN obtains accurate count estimation, the quality of density graph is poor(e.g., density map in Fig. 4) so that adversely affect other higher level cognition tasks which depend on them. Again, our MPC is more effective(less training time and single-column structure) than MCNN but achieves worse MAE (1.12), since the largest object is only about 30 pixels tall on UCSD dataset, this limits the performance gain achieved by Front-end from leveraging large context information. Also, result is shown in Table 8 which indicates our model performs stably on this dataset.

4.5 Training data and runtime speed

Here we consider whether there is an impact on the final results when using an augmented training set. Next we report runtime speed of using the four most competitive

Table 8 The standard deviation of absolute error in four datasets

Dataset	Std_AE
ShanghaiTech Part A [18]	79.4
ShanghaiTech Part B [18]	12.8
UCF_CC_50 [55]	256.3
The UCSD [6]	1.07

methods on the tested datasets. The experiment was run on the ShanghaiTech Part B dataset whose resolution was fixed relative to the Part A and UCF_CC_50 datasets and higher than the UCSD dataset.

Data augmentation A commonly used strategy to improve the performance of CNNs is to augment training data. Here, we rotate the images to 8 different angles and crop the rotated images into four quarters without overlapping; we also flipped the patches right-left-wise and added small random Gaussian noise. But we noted there was no obvious improvement, and thus it did not necessarily require data augmentation. This is most likely because we extracted all the possible image patches to train the network so that it already covers many useful permutations of the input.

Runtime speed Overall, MCNN [18], Switch-CNN [20], CP-CNN [45] and our proposed method(MPC) perform the best on the tested datasets. We summarize their runtime speed in Table 9. The runtime is tested with caffe on GeForce GTX 1080 on 1024 × 768 images of ShanghaiTech Part B. Both Switch-CNN and CP-CNN use pre-trained VGG model as density level classifier which spend a large portion of parameters on categorizing the input images into various density levels and do not run in a fully convolutional fashion, resulting in slower prediction. CP-CNN is especially slow because it also uses a 64 × 64 sliding window classifier for per-pixel prediction to get the local context information. In contrast, MCNN [18] is the fastest among these methods since it uses much smaller models and does not need extra data to pre-train the model. Our MPC also runs fast because it uses a single column CNNs structure with pure convolutional layers as the backbone.

Table 9 The runtime of the four most competitive methods

	FPS
MCNN [18]	40.5
Switching-CNN [20]	1.72
CP-CNN [45]	0.02
MPC(ours)	33.5

5 Conclusion

As CNNs are becoming the leading choice for crowd counting, the biggest concern of crowd counting with this technique is the amount of contextual information output. Most of the work has adopted multi-scale or multi-task learning CNN architectures in order to counteract this issue, but the designs they used also introduce two significant disadvantages: large amount of training time and complex structure. We decided, however, to rethink simply-nested CNNs architectures.

In this paper, we propose MPC which generates high-quality density map end-to-end. By concatenating multiple feature maps of different scales (and thus different levels of contextual information) in the congested scenes, MPC network can expand the receptive field without losing resolution, and thus make a tradeoff between recognition and resolution. The proposed method is also influenced by diverse deep supervision (DDS) which guide the MPC network predictions. Compared with the general supervision, DDS takes advantage of all of the intermediate layers or hierarchical diversities within the network. It can incrementally improve the strength of the supervision and be well-suited the network coarse-to-fine feature extraction paradigm. Consequently, our proposed MPC produces high-quality density map, meanwhile achieves better counting results on three crowd datesets than MCNN and Switch-CNN. Although CP-CNN has better results on some datasets than our proposed method, it is much slower since it uses both a bloated network structure and sliding window per-pixel prediction. Taking into consideration of model size and running time, our method is the most favorable, especially for cases requiring real-time prediction. Ideally, we would wish to train one crowd counter that is able to perform well on multiple benchmarks. Future works will focus on improving generalization capability of network and adding perspective information into the network training.

Finally, our idea is not limited in crowd counting tasks and expected to be extended to counting tasks in other fields such as cell microscopy, vehicle counting, environmental survey, etc.

Appendix: Supplementary Material

This section presents some additional results of MPC for the three datasets (Shanghai Tech [18], UCF_CC_50 dataset [55] and UCSD dataset [6].The PSNR (Peak Signal-to-Noise Ratio) and the SSIM (Structural Similarity in Image) perform to evaluate quality of generated density maps. Results on sample images from these datasets are shown in Figs. 6, 7, 8 and 9, which represent a variety of density levels.

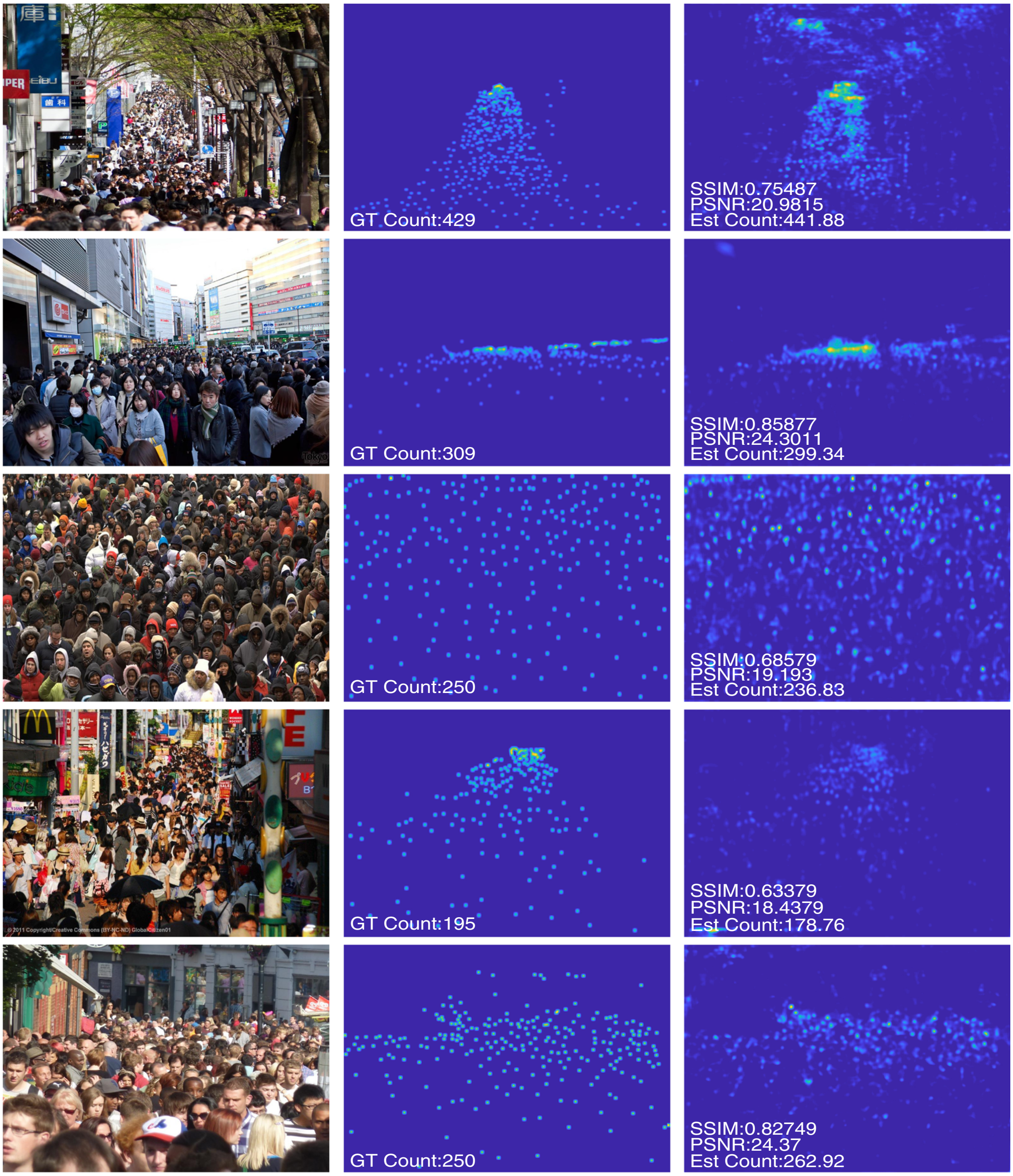

Fig. 6 Results of our MPC medel on Shanghai Tech Part A dataset [18]. Left column: Input images. Middle column: Ground truth density maps. Right column: Estimated density maps

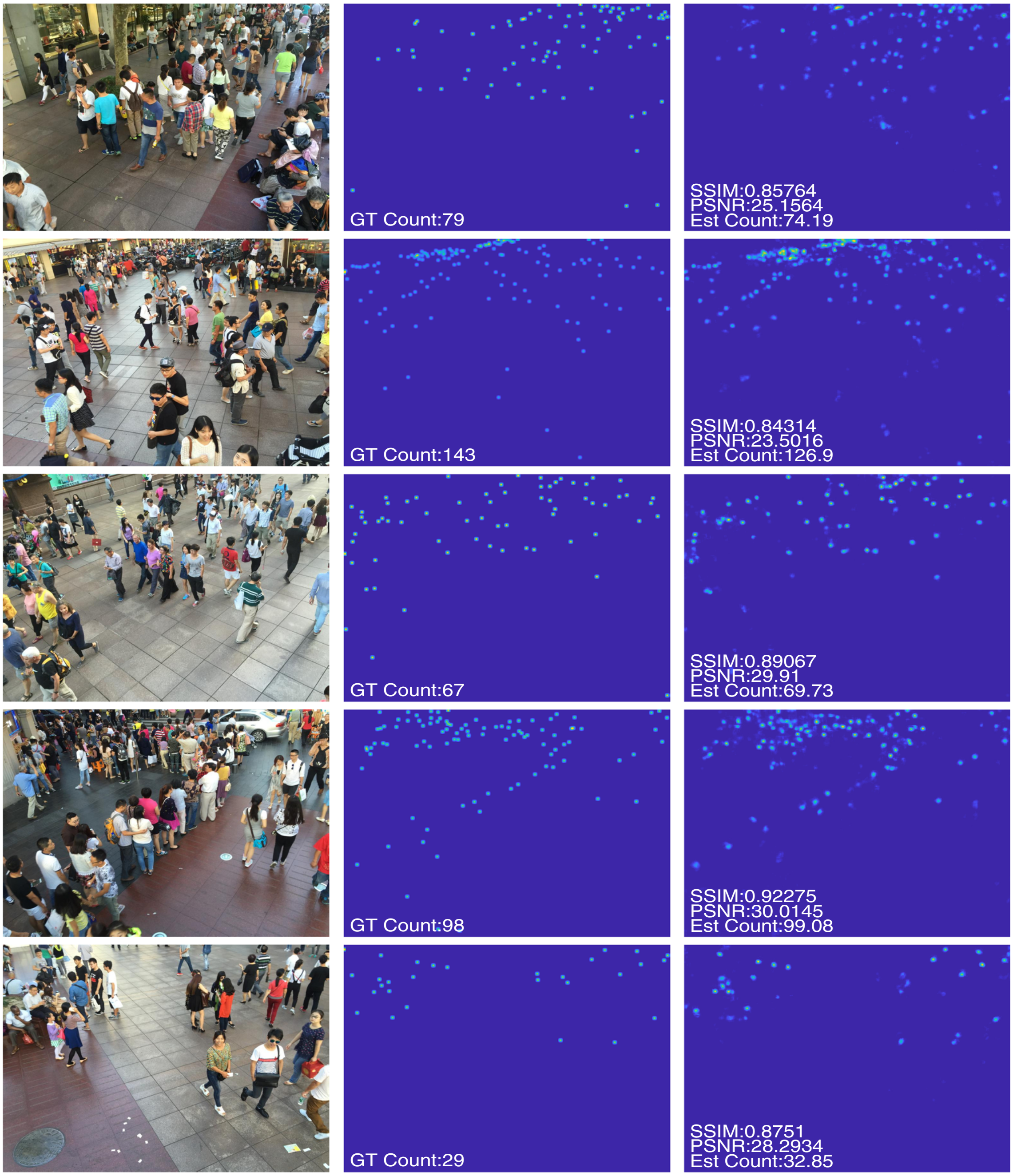

Fig. 7 Results of our MPC medel on Shanghai Tech Part B dataset [18]. Left column: Input images. Middle column: Ground truth density maps. Right column: Estimated density maps

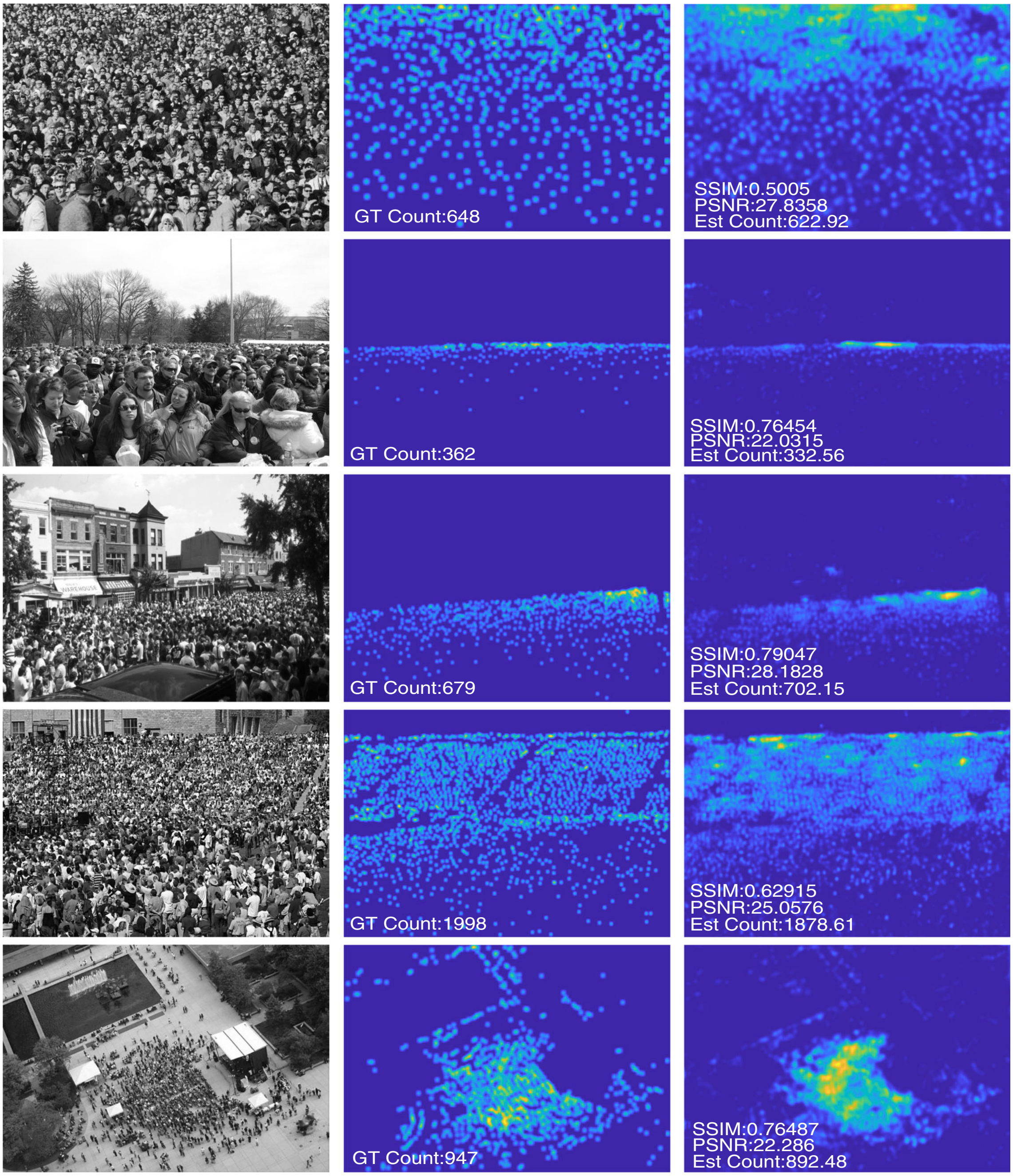

Fig. 8 Results of our MPC medel on UCF_CC_50 dataset [55]. Left column: Input images. Middle column: Ground truth density maps. Right column: Estimated density maps

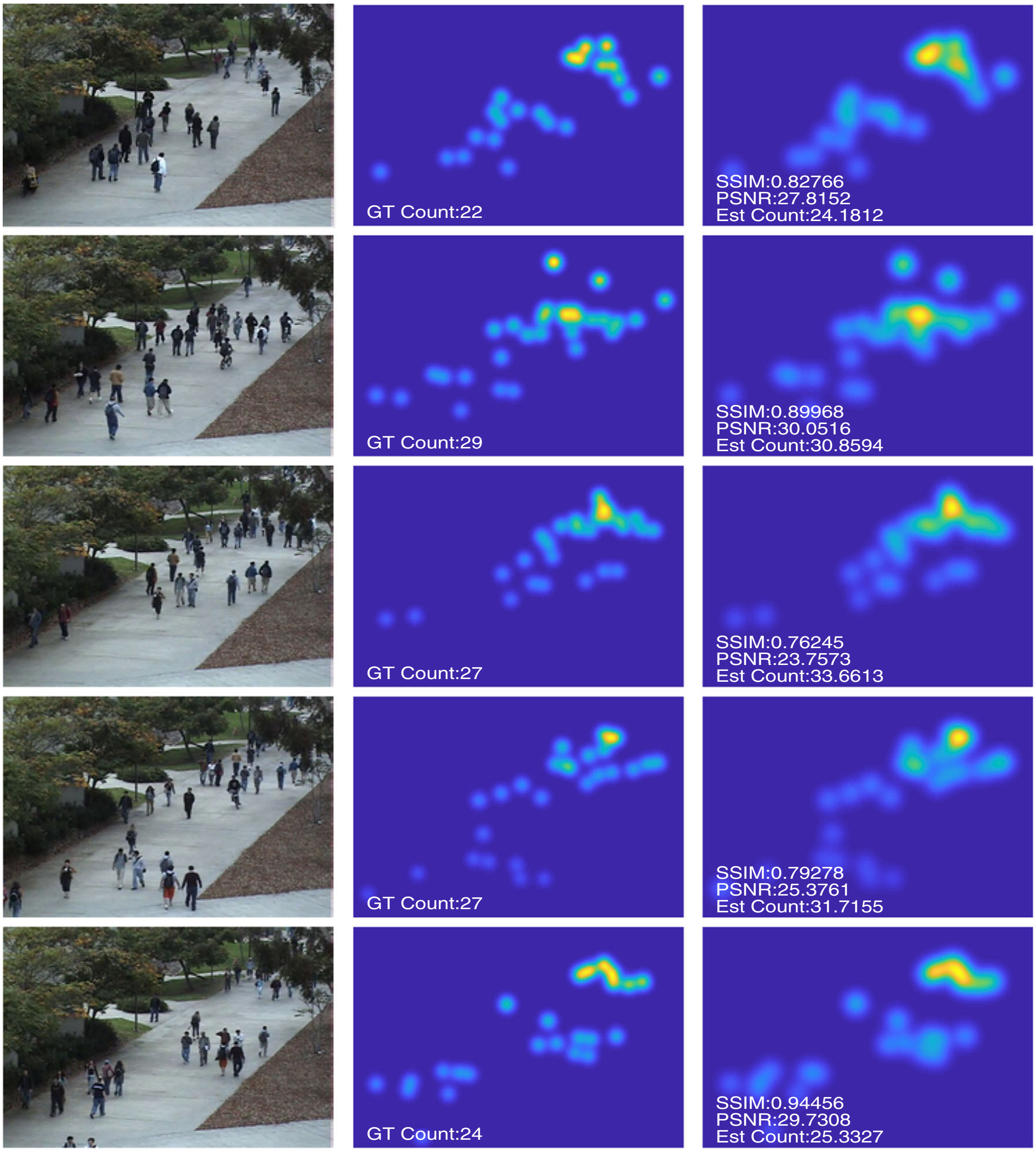

Fig. 9 Results of the our MPC model on UCSD dataset [6]. Left column: Input images. Middle column: Ground truth density maps. Right column: Estimated density maps

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

References

1. Leibe B, Seemann E, Schiele B (2005) Pedestrian detection in crowded scenes. IEEE Conf Comput Vis Pattern Recogn 1:875–885
2. Zhao T, Nevatia R (2003) Bayesian human segmentation in crowded situations. IEEE Conf Comput Vis Pattern Recogn 2:459–466
3. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. IEEE Comput Soc Conf Comput Vis Pattern Recogn 1:886–893
4. Hou Y-L, Pang GK (2011) People counting and human detection in a challenging situation. IEEE Trans Syst Man Cybern-Part Syst Hum 41(1):24-33. 13
5. Ryan D, Denman S, Fookes C, Sridharan S (2009) Crowd counting using multiple local features. Digital Image Computing: Techniques and Applications(DICTA), pp 81–88
6. Chan AB, Liang Z-SJ, Vasconcelos N (2008) Privacy preserving crowd monitoring: Counting people without people models or

tracking. The IEEE conference on computer vision and pattern recognition(CVPR), pp 1–7

7. Marana A, daFontoura.Costa L, Lotufo R, Velastin S (1999) Estimating crowd density with Minkowski fractal dimension. Proc IEEE Int Conf Acoust Speech Signal Process 6:3521–3524
8. Davies AC, Yin JH, Velastin S (1995) Crowd monitoring using image processing. Electron Commun Eng J 7(1):37–47
9. Paragios N, Ramesh V (2001) A MRF-based approach for real-time subway monitoring. IEEE Comput Soc Conf Comput Vis Pattern Recogn(CVPR) 1:I–1034
10. Rahmalan H, Nixon MS, Carter JN (2006) On crowd density estimation for surveillance. The Institution of Engineering and Technology Conferenceon Crime and Security, pp 540–545
11. Kong D, Gray D, Tao H (2005) Counting pedestrians in crowds using view point invariant training. In: Proceedings of British Machine Vision Conference(BMVC)
12. Lempitsky V, Zisserman A (2010) Learning to count objects in images. In: Advances in Neural Information Processing Systems, pp 1324–1332
13. Fiaschi L, Nair R, Koethe U, Hamprecht FA (2012) Learning to count with regression forest and structured labels. In: ICPR, pp 2685–2688
14. Pham VQ, Kozakaya T, Yamaguchi O, Okada R (2015) Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In: Proceedings of the IEEE International Conference on Computer Vision(CVPR), pp 3253–3261
15. Wang Y, Zou Y (2016) Fast visual object counting via example-based density estimation. In: IEEE international conference on image processing (ICIP), pp 3653–3657. https://doi.org/10.1109/ICIP.2016.7533041
16. Wang C, Zhang H, Yang L, Liu S, Cao X (2015) Deep people counting in extremely dense crowds. In: Proceedings of the 23rd ACM international conference on Multimedia, pp 1299–1302
17. Fu M, Xu P, Li X, Liu Q, Ye M, Zhu C (2015) Fast crowd density estimation with convolutional neural networks. Eng Appl Artif Intell 43:81–88
18. Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. IEEE conference on computer vision and pattern recognition(CVPR)
19. Shang C, Ai H, Bai B (2016) End-to-end crowd counting via joint learning local and global count. 2016 IEEE International Conference on Image Processing(ICIP), pp 1215–1219
20. 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(CVPR)
21. Onoro-Rubio D, Lopez-Sastre RJ (2016) Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision(ECCV), pp 615–629
22. Walach E, Wolf L (2016) Learning to count with cnn boosting. In: European Conference on Computer Vision(ECCV), pp 660–676
23. Hu P, Ramanan D (2016) Finding Tiny Faces. arXiv:1612.04402
24. Yu F, Koltun V (2016) Multi-Scale Context aggregation by dilated convolutions. ICLR
25. Badrinarayanan V, Handa A, Cipolla R (2017) SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixelwise labelling. IEEE Trans Pattern Anal Mach Intell 39:2481–2495
26. Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation. In: the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3431–3440
27. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention(MICCAI), pp 234–241
28. Lee C, Xie S, Gallagher P, Zhang Z, Tu Z (2015) Deeply supervised nets. In: AISTATS
29. Kong D, Gray D, Tao H (2006) A Viewpoint Invariant Approach for Crowd Counting. In: The 18th International Conference on Pattern Recognition(ICPR), pp 1187–1190
30. Chan AB, Morrow M, Vasconcelos N (2009) Analysis of crowded scenes using holistic properties, in Performance Evaluation of Tracking and Surveillance Workshop at CVPR, pp 31–37
31. Shimosaka M, Masuda S, Fukui R, Moriand T, Sato T (2011) Counting pedestrians in crowded scenes with efficient sparse learning. In: First Asian Conference on Pattern Recognition (ACPR), pp. 27-31
32. Khan U, Klette R (2016) Logarithmically improved property regression for crowd counting. Pacific-Rim Symposium on Image and Video Technology:Image and Video Technology, pp 123–135
33. Chan AB, Vasconcelos N (2009) Bayesian poisson regression for crowd counting. In: 2009 IEEE 12th International Conference on Computer Vision, pp 545–551
34. Chen K, Loy CC, Gong S, Xiang T (2012) Feature mining for localised crowd counting. Inproceedings British Machine Vision Conference, pp 21.1–21.11
35. Marana A, Costa LdF, Lotufo R, Velastin S (1998) On the Efficacy of Texture Analysis for Crowd Monitoring. In: 1998. Proceedings. SIBGRAPI'98. International Symposium on Computer Graphics, Image Processing, and Vision, pp 354–361
36. Fradi H, Dugelay JL (2012) People counting system in crowded scenes based on feature regression. In: Proceedings of European Signal Processing Conference, pp 27–31
37. Kumagai S, Hotta K, Kurita T (2017) Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting. arXiv:1703.09393
38. Marsden M, McGuiness K, Little S, E.O'Connor N (2016) Fully convolutional crowd counting on highly congested scenes. arXiv:1612.00220
39. Sheng B, Shen C, Lin G, Li J, Yang W, Sun C (2016) Crowd counting via weighted VLAD on dense attribute feature maps. IEEE Transactions on Circuits and Systems for Video Technology
40. Di K, Ma Z, Chan AB (2017) Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks-Counting, Detection, and Tracking. preprint arXiv:1705.10118
41. Arteta C, Lempitsky V, Zisserman A (2016) Counting in the wild. In: European Conference on Computer Vision. Springer, pp 483–498
42. Zhao Z, Li H, Zhao R, Wang X (2016) Crossing-line crowd counting with two-phase deep neural networks. In: European Conference on Computer Vision. Springer, pp 712C726
43. Sindagi VA, Patel VM (2017) Cnn-based cascaded multitask learning of high-level prior and density estimation for crowd counting. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)
44. Zhang C, Li H, Wang X, Yang X (2015) Cross-scene crowd counting via deep convolutional neural networks. IEEE conference on computer vision and pattern recognition(CVPR), pp 833–841
45. Sindagi VA, Patel VM (2017) Generating High-Quality Crowd Density Maps using Contextual Pyramid CNNs. IEEE International Conference on Computer Vision (ICCV)
46. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the Inception Architecture for Computer Vision. IEEE conference on computer vision and pattern recognition(CVPR)
47. Boominathan L, Kruthiventi SS, Babu RV (2016) Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 2016 ACM on Multimedia Conference, ACM, pp 640–644
48. Simonyan K, Zisserman A Very deep convolutional networks for large-scale image recognition. In: ICLR, 2015

49. Girshick R (2015) Fast R-CNN. In: IEEE ICCV, pp 1440–1448
50. Yang J, Price B, Cohen S, Lee H, Yang M-H (2016) Object contour detection with a fully convolutional encoder-decoder network. arXiv:1603.04530
51. Shi M, Caesar H, Ferrari V (2018) Crowd counting via scale-adaptive convolutional neural network. IEEE Winter Conference on Applications of Computer Vision (WACV)
52. Dubrovina A, Kisilev P, Ginsburg B, Hashoul S, Kimmel R (2016) Computational mammography using deep neural networks. In: Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization, pp 1–5
53. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2016) DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Preprint: arXiv:1606.00915
54. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. (2014) Caffe: Convolutional architecture for fast feature embedding. In: ACM MM, pp 675–678
55. Idrees H, Saleemi I, Seibert C, Shah M (2013) Multisource multi-scale counting in extremely dense crowd images. IEEE conference on computer vision and pattern recognition (CVPR), pp 2547–2554
56. Casella G, Berger R (1990) Statistical inference, 2nd edn. Duxbury Press, p 686
57. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

Haiying Jiang received the B.Eng. and M.Sc. From Southwest Jiaotong University in 2006 and 2009, respectively. She is currently working towards the PhD degree in Control Science & Engineering at Southwest Jiaotong University. Her research interests include computer vision, crowd counting and deep learning.

Weidong Jin received the B.S. degree, the M.S. degree and the Ph.D. degree in automatic control from the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 1982, 1990 and 1998 respectively. He is currently a Professor in the School of Electrical Engineering, Southwest Jiaotong University. His research interests include intelligent information processing, pattern recognition, and optimization and simulation.