Applied Intelligence (2018) 48:3360–3371
https://doi.org/10.1007/s10489-018-1150-1

Skip-connection convolutional neural network for still image crowd counting

Luyang Wang[1] · **Baoqun Yin**[1] · **Aixin Guo**[1] · **Hao Ma**[1] · **Jie Cao**[1]

Published online: 23 February 2018

Abstract
In recent years, crowd counting in still images has attracted many research interests due to its applications in public safety. However, it remains a challenging task for reasons of perspective and scale variations. In this paper, we propose an effective Skip-connection Convolutional Neural Network (SCNN) for crowd counting to overcome the issue of scale variations. The proposed SCNN architecture consists of several multi-scale units to extract multi-scale features. Each multi-scale unit including three convolutional layers builds connections between the input and each convolutional layer. In addition, we propose a scale-related training method to improve the accuracy and robustness of crowd counting. We evaluate our method on three crowd counting benchmarks. Experimental results verify the efficiency of the proposed method, and it achieves superior performance compared with other methods.

Keywords Crowd counting · Convolutional neural network · Multi-scale unit · Scale-related training method

1 Introduction

Crowd counting is a visual cognitive task which aims at accurately estimating the number of people in a crowded scene. It has become an important topic in the field of computer vision due to various potential practical applications such as public safety management, crowd control and video surveillance. However, existing crowd counting methods are far from optimal owing to the following difficulties. The distribution of crowds across the scenes is diverse. Crowding and occlusions among people are common characteristics of the dense crowd scenes. In addition, scale variations of crowd scenes are of variety, as the camera viewpoint transformation results in perspective effects. Due to the presence of these complexities, crowd counting is still challenging.

One key challenge to fix these complexities is large scale variations due to perspective effects. To solve the above issue, various crowd counting methods have been put forward. Early methods [1] utilized hand-crafted features and head detection failed in dense crowd scenes. Recently, inspired by the success of CNNs on various vision tasks, many multi-scale CNN architectures [2–4] for crowd counting have achieved remarkable performance improvements. These methods generally tackled the issue of scale variations via multi-column networks to estimate the density maps of still crowd images. However, a major limit of multi-column networks is that each column with different filter sizes only works for a single scale. Due to the large memory consumption, multi-column networks usually have two or three columns, which can only extract few multi-scale features. In principle, multi-column networks cater scale variations by increasing receptive fields of different sizes in the network.

Considering the above observations, we aim at learning a model to cope with large scale variations by increasing the number of receptive fields of different sizes as much as possible. Motivated by the work [5] for semantic

✉ Luyang Wang
ly1105@mail.ustc.edu.cn

Baoqun Yin
bqyin@ustc.edu.cn

Aixin Guo
guoaixin@mail.ustc.edu.cn

Hao Ma
tmac01@mail.ustc.edu.cn

Jie Cao
stud2012@163.com

[1] Department of Automation, University of Science and Technology of China, Hefei, China

segmentation, we propose a Skip-connection Convolutional Neural Network to estimate the density maps of input images. In their work, they built two skip-connections from low layers to high layers, which fused three kinds of scale features. Our SCNN cascades four multi-scale units. Each multi-scale unit consists of three convolutional layers and we build extra skip-connections from the input of the multi-scale unit to each convolutional layer. Compared with the multi-column network, the SCNN consumes less memory and contains more receptive fields of different sizes due to the reuse of low layer features. In addition, the skip-connections on the entire network only increase the same number of receptive field sizes as the number of connections. However, the number of receptive fields increases exponentially in our unit style network architecture.

In order to accurately estimate the count of people in a still image, we also provide a novel scale-related training method that works as an auxiliary means to tackle the issue of scale variations. Inspired by the multi-scale input network [6], we use the input images of two scales during training. The difference between our method and the multi-scale input network is that the training images of each scale are applied to train a model individually instead of training the model with multi-scale images simultaneously. The training images of large scale are used to train the model at first, and then we fine-tuning the weight parameters with the images of the original scale.

The contributions of this paper can be summarized as follows:

1. We present a Skip-connection Convolutional Neural Network for crowd counting in still images. Based on the idea of increasing receptive fields of different sizes, we first propose the multi-scale unit formed SCNN to overcome scale variations and perspective. The multi-scale unit builds connections between the input and each layer of the multi-scale unit.
2. A scale-related training method is proposed to improve overall counting performance. Different from traditional training methods, the images of the original size are used to fine-tune the training result of large size images.
3. We evaluate our network architecture on UCF_CC_50 [1], ShanghaiTech [2], and compare with the state-of-the-art technologies. The results confirm the effectiveness of our method.

The remainder of this paper is organized as follows. Section 2 surveys related works on crowd counting. Section 3 introduces detailed SCNN architecture for crowd counting and analyzes the receptive field size. Section 4 presents the descriptions of the proposed scale-related training method. Section 5 presents the experimental results and discussions. Section 6 concludes the paper and the future work.

2 Related work

In this section, we mainly discuss the crowd counting methods proposed in the existing literature. We also introduce several related works of the multi-scale convolutional neural network architecture, as we design the SCNN from the viewpoint of extracting multi-scale features.

Crowd counting as a computer vision task has been tackled by a number of methods. These methods can be roughly divided into three categories: detection-based counting, regression-based counting and CNN-based counting.

Detection-based counting. Early methods of crowd counting [7–11] generally adopted a visual object detector to scan individuals over frames of a video or still images, and the result of counting is the sum of the detected individuals. Many kinds of detectors have been employed to detect individuals in an image. Lin et al. [7] utilized the Haar wavelet transform to detect the feature area of the head-like contour. In the literature [12], Li proposed the head-shoulder detection with a foreground segmentation framework. Wu and Nevatia [8] introduced edgelet features to learn human body part detectors with a boosting method, and improved the robustness of individual overlaps. However, detection-based counting is limited by occlusions between people in a crowded scene. As the crowd becomes dense, detection performance drops rapidly.

Regression-based counting. To overcome the bottleneck of detection-based methods in the dense crowds, regression-based counting [13–17] is intended to establish a map between low-level features and the number of people, instead of detecting individuals in a crowd scene. These methods first extract features from a crowd region, and then predict the crowd count by training a regression model. Various features have been employed such as textures [14, 18] and edge information [17, 19]. Common regression models include linear regression [20], piecewise linear regression [21], Bayesian regression [16, 19], ridge regression [14] and Gaussian process regression [13]. Regression-based counting is effective for tackling the problem of occlusions, while it only gives the global counts of people and ignores the crowd spatial distribution information. Lempitsky and Zisserman et al. [15] put forward to estimate an image density map whose integral over an image region gave the count of objects within that region. The density map is employed by a variety of recent methods, as it describes the crowd spatial distribution and is easy to calculate the count of people.

CNN-based counting. Recently, CNN-based methods have achieved great success in various vision tasks such as object detection [22], classification [23] and semantic segmentation [5]. Many CNN architectures [24–26] have also been applied to crowd counting and improved the counting accuracy. Zhang et al. [24] considered cross-scene crowd counting and presented a CNN trained alternatively with two loss functions. However, their method requires perspective maps during training and testing, which is not accessible in the practical applications of crowd counting. Several multi-task learning methods [25–27] have been proposed, which developed auxiliary tasks to improve the counting performance. These methods, however, fail to consider scale variations that are commonly found in crowd images. Zhang et al. [2] proposed a multi-column fully convolutional network (FCN) to extract multi-scale features. The multi-column FCN consists of three columns with different convolution kernel sizes. Boominathan et al. [4] adopted a combination of deep and shallow FCN to predict the density map for a given crowd image. These multi-column methods partially improve the issue of scale variations, but their network architectures are of much complexity and only extract few multi-scale features. In this work, we focus on the issue of scale variations of crowd counting. We develop a simple but effective multi-scale unit by taking the input features combined with the output of the current convolutional layer as the input of the next convolutional layer. Our proposed single column SCNN is built by cascading multi-scale units to recover rich scale information from images.

Multi-scale CNN architecture. In recent years, various methods demonstrate astonishing results in pixel-level visual tasks. Among these models, one of the key elements to success is the use of multi-scale features [28]. There are primarily three types of multi-scale network architectures in the CNN: multi-column network [29], skip-net [30, 31] and multi-scale input [32]. The multi-column network is illustrated in Fig. 1a. Input data are fed into multiple columns, and the output data of each parallel column are concatenated as the final output. [2, 4] are typical multi-column network architectures for crowd counting. As Fig. 1b illustrated, the skip-net builds a connection between low-level features and high-level output. Therefore, the features of different levels are combined and fed into an output layer. One common ground of these two structures is that multi-scale features are obtained by increasing receptive fields of different sizes. The illustration of multi-scale input is shown in Fig. 1c. The input images are resized to several scales to train a single network. The pyramid network is a widely used example of multi-scale input [32].

In this work, the design of our SCNN is based on the idea of skip-net. However, the most significant difference with skip-net is that our architecture builds multiple connections between the input and each layer of the multi-scale unit rather than just one or two skip-connections between low layers and high layers. The purpose of this design is to increase more receptive fields of different sizes, further to extract more multi-scale features.

3 Skip-connection CNN architecture for crowd counting

As discussed in Section 1, existing convolutional architectures deal with the issue of scale variations by using multi-column CNN architectures which involve receptive fields of different sizes. We think it is important to increase the number of receptive fields of different sizes in the network to reduce the influence of scale variations. Therefore, we propose the SCNN architecture that consists of four multi-scale units to estimate the density maps of input images. In

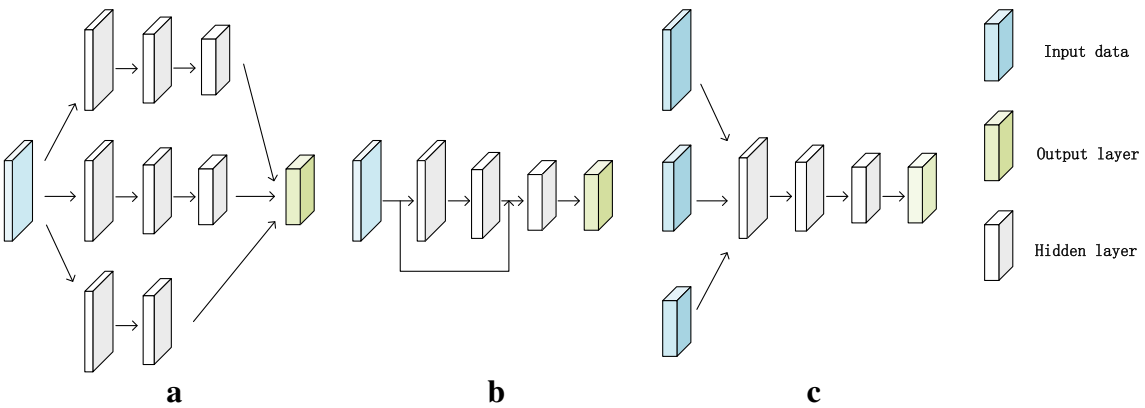

Fig. 1 Illustration of different multi-scale deep learning architecture: **a** multi-column network, **b** skip-net, **c** multi-scale input

this section, we first introduce the density map which is the regression result of our SCNN. Then the multi-scale unit and the proposed SCNN are described in detail. Finally, we analyze the receptive field sizes in the network and discuss the principle that our SCNN involves more receptive fields of different sizes.

3.1 Density map

The density map and the number of people are two common outputs of the CNN for crowd counting in a still image. Compared with the number of people, the density map provides more information about the distribution of crowds. The number of people over the entire input image can be obtained by integrating the density map. The quality of density maps in the training set is a major factor affecting the performance of crowd counting, since the CNN is trained to estimate the density map of a training image. Following the previous work [15], we generate the ground truth density map of a crowd image with labeled heads of people in the training set. For a given labeled crowd image, a head annotation at pixel x_i can be formalized as a unit impulse function $\delta(x - x_i)$, where x represents the two-dimensional image coordinates. Therefore, an image with N heads can be represented as the following formula. Figure 2 displays a crowd image and its density map.

$$H(x) = \sum_{i=1}^{N} \delta(x - x_i) \tag{1}$$

Convolve $H(x)$ with a Gaussian kernel G_σ, we can obtain the density map $F(x) = H(x) * G_\sigma(x)$. Zhang et al. [2] proposed an adaptive Gaussian kernel G_{σ_i} due to perspective. The spread parameter σ_i of the adaptive Gaussian kernel is determined based on the average distance $\bar{d}_i$ from the k nearest head annotation pixels to x_i, instead of a constant σ. The density map with the adaptive Gaussian kernel can be calculated by formula (2), and empirically $\beta = 0.3$.

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_{\sigma_i}(x), \ \sigma_i = \beta \times \bar{d}_i \tag{2}$$

3.2 Skip-connection CNN architecture

It is common sense that the heads of people in the distance are quite smaller than those located nearby in a still image. The difference of head sizes caused by perspective leads to different scales in crowd images. It is difficult to extract multi-scale crowd features by CNNs with the receptive field of a single size. Therefore, it is necessary to design a CNN architecture with receptive fields of different sizes to estimate density maps. Motivated by the skip-net, we propose the SCNN which is mainly composed of several multi-scale units to learn the map between input images and corresponding density maps.

Multi-scale unit. Consider a feature map f_0 as the input of the multi-scale unit, and the output of the i^{th} layer is f_i. We denote the non-linear transformation of the i^{th} layer as $T_i()$. For the traditional CNN architecture, the output of the i^{th} layer is the input of the $(i+1)^{th}$ layer, which can be described as follows: $f_{i+1} = T_{i+1}(f_i)$. However, our multi-scale unit adds skip-connections between input data and each convolutional layer.

As illustrated in Fig. 3, the multi-scale unit consists of three convolutional layers and the architecture from the input feature maps to the output can be formulated as

$$\begin{cases} f_1 = T_1(f_0) \\ f_2 = T_2([f_0, f_1]) \\ f_3 = T_3([f_0, f_2]) \end{cases} \tag{3}$$

where $[f_0, f_i]$ denotes the concatenation of the feature maps. In our multi-scale unit, the input of each convolutional layer includes the input data of the multi-scale unit and the output of the preceding layer except the first convolutional layer. The feature maps of the last convolutional layer are equally the output of the multi-scale unit. Rectified linear unit (ReLU) is applied as the activation function followed by each convolutional layer [33]. $T_i()$ is a composite function of convolution and ReLU. Convolutional layers in each multi-scale unit of

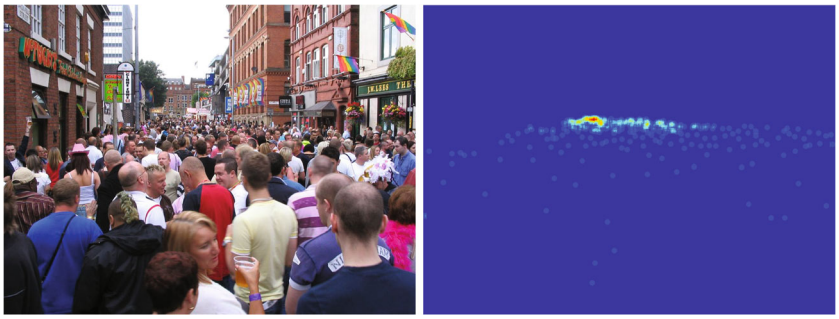

Fig. 2 An input image and the corresponding density map obtained by convolving Gaussian kernels

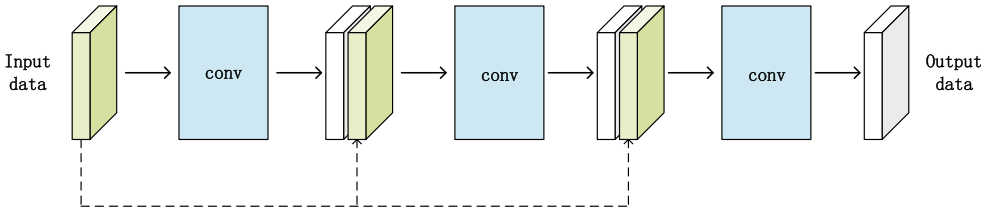

Fig. 3 The architecture of the proposed multi-scale unit

the proposed SCNN have the same size of kernels and number of feature maps.

SCNN architecture. Figure 4 shows the overall architecture of the proposed SCNN. It contains three parts. The first part is a traditional convolutional layer with 64 9 × 9 convolution kernels. The second part consists of 4 multi-scale units, and each convolutional layer in a multi-scale unit has the same size kernels. Each multi-scale unit adds two skip-connections and covers three kinds of scales. The number of receptive fields of different sizes brings about a significant increase by cascading multi-scale units in the SCNN. Convolutional layers in the first two multi-scale units have 32 7 × 7 kernels. The third multi-scale unit has 16 7 × 7 kernels for each convolutional layer, and there are 16 5×5 kernels for convolutional layers of the last multi-scale unit. The last part contains two convolutional layers, which aim at transforming multi-scale features to the final density map. The two convolutional layers have 128 and 1 kernels of 1 × 1 respectively. Max pooling is applied for each 2×2 region after the first convolutional layer and the first multi-scale unit respectively. ReLu is also used after each convolutional layer except the last one of 1 × 1.

Our SCNN takes an arbitrary size crowd image as input and outputs the corresponding density map. The spatial resolution of the estimated density map is 1/4 of the input crowd image due to the max pooling. Therefore, it is necessary to down-sample the ground truth density maps of the training images during the training stage. We choose the Euclidean distance to assess the difference between the estimated density map and the corresponding ground truth. The loss function is formulated as follows:

$$L(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|F_i - F(X_i;\Theta)\|^2 \tag{4}$$

where $L(\Theta)$ denotes Euclidean loss. Θ represents the weight parameters to be optimized in the network. N is the number of training images, and X_i is the i^{th} input image. F_i and $F(X_i;\Theta)$ are the i^{th} ground truth density map and the estimated density map respectively. We use the stochastic gradient descent (SGD) and backpropagation to minimize the loss function.

3.3 Analysis of receptive field size

As mentioned above, increasing receptive fields of different sizes in the network is an effective means to overcome the issue of scale variations for still image crowd counting. In the CNN architecture, the receptive fields of different sizes are likely to capture characteristics of crowds at different scales. Therefore, it is natural to use receptive fields of multiple sizes in a CNN architecture to suppress the perspective distortion and scale variations.

In [34], Google proposed their third version of the Inception architecture. They presented that a 5 × 5 convolution can be replaced by two-layer 3 × 3 convolutions with the same input size and output depth (as shown in Fig. 5). So, we think that the convolutions with kernels of different sizes in a traditional single column CNN can be viewed as a receptive field of a single size. In [35], the

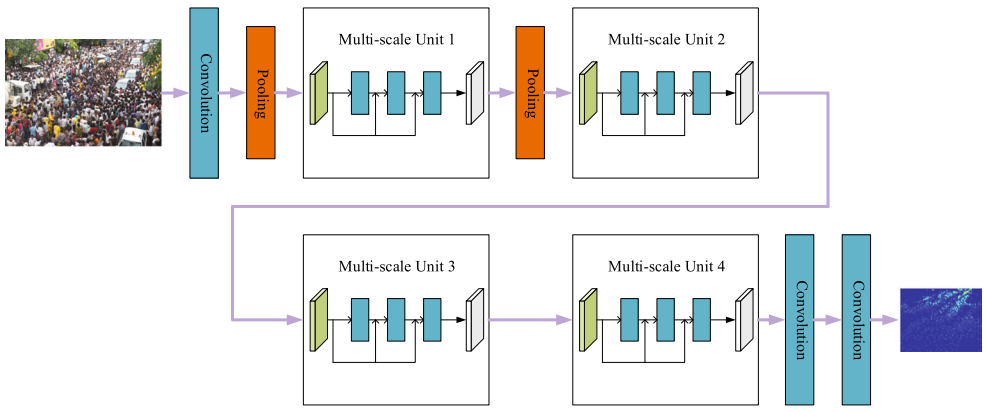

Fig. 4 The architecture of Skip-connection Convolutional Neural Network for crowd counting. In each multi-scale unit, the green and white cubes are input and output feature maps respectively. The blue rectangle in the multi-scale unit represents the convolutional layer

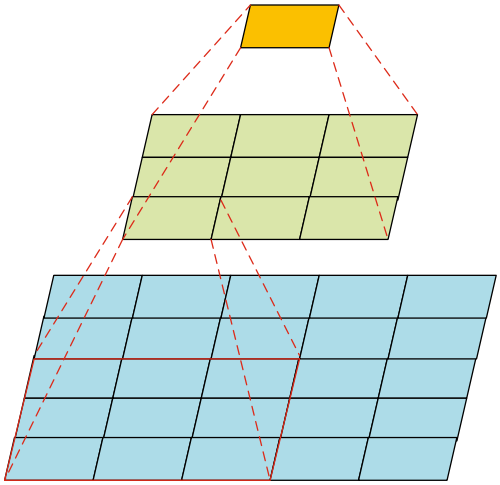

Fig. 5 Illustration of two 3 × 3 convolutional layers replacing a 5 × 5 convolutional layer

author provided the calculation formula of the receptive field size as follows:

$$j_{n+1} = j_n \times s \tag{5}$$

$$r_{n+1} = r_n + (k-1) \times j_n \tag{6}$$

where r_n is the receptive field size upto the n^{th} layer. k is the convolution kernel size of the current layer. s is the convolution stride size. j_n is the distance between two adjacent features of the n^{th} layer. The initial values of j_0 and r_0 are both 1. Receptive field sizes of CNNs can be calculated by iterating (6).

Existing multi-column crowd counting methods extract features of a scale from each column. Multi-column networks usually consist of two or three columns, as too many layers will lead to large memory consumption. As a result, there are only two or three receptive fields of different sizes in the network. Take MCNN [2] as an example, it only contains three sizes of receptive fields in three parallel columns and the receptive field sizes are 48, 34 and 20 respectively. However, our SCNN overcomes this limitation by the reuse of low-level features. Each skip-connection adds an extra receptive field without introducing a CNN column. Considering our proposed multi-scale unit, it is composed of three convolutional layers and the input of the multi-scale unit combined with the output of each convolutional layer works as the input of the next convolutional layer. There are three sizes of receptive fields in a multi-scale unit. Suppose the convolution kernel size in the multi-scale unit is 7 × 7 which is used in our SCNN. The calculation results of the receptive field sizes are 19, 13 and 7. Our proposed SCNN architecture cascades four multi-scale units. This design is more efficient than skip-connections across the entire network (i.e. skip net). Our unit style network architecture makes the number of receptive field of different sizes increase exponentially. For example, cascading two multi-scale units of different convolution kernel sizes can produce nine sizes of receptive fields. In our SCNN, the convolution kernel size of the first three multi-scale units is 7×7, and another one is 5×5. Due to the same convolution kernel size and repetition, there are 33 kinds of receptive field sizes in the SCNN. The largest size of the receptive field in the SCNN is 240, and the smallest size is 84.

The analysis of the receptive field size demonstrates that our proposed SCNN contains more receptive fields than the previous architectures. The gap between the largest size of the receptive field and the smallest one is quite large, and it is adapted to crowd scenes of diverse scales. Compared with other multi-column architectures, our proposed architecture can extract more multi-scale features with fewer convolutions, and deepen the network.

4 Scale-related training method

In this section, we propose a novel scale-related training method to improve the performance of complex scene crowd counting. Our training method augments the training set with two scales of training samples, and trains the SCNN in two stages. As we know, the deep convolutional neural network is a data-driven technology, which requires a large amount of data to train the network parameters. The amount of data directly influences the performance of the network. However, existing crowd counting datasets have only a few hundred images and they are not sufficient for training the CNN. Therefore, the dataset augmentation scheme is significant to increase the number of training samples. Cropping patches with overlap from each training image is a common training set augmentation technology. However, Marsden et al. [36] found that the pixel-level tasks can potentially overfit when the patches overlap. They proposed a training set augmentation scheme to ensure there is no redundancy.

Following their work, we propose the scale-related training method. The augmentation scheme without redundancy may reduce training samples because of the limitation of patches without overlap. In order to involve more scale information in the training set and increase training samples,

we employ the augmentation scheme without redundancy to crop two scales of patches from each training image. Then the training samples of the two scales are adopted to train the CNN, individually. The method is described in detail below. We first resize the spatial resolution of training images to 1.5 times of the original size and evenly crop each image into 9 patches without overlap. After screening out the patches that have no people, all the remaining patches as well as their horizontal flips constitute the first training set. Next, we crop 4 patches from each training image of the original size, and each patch is 1/4 size of the image. The patches together with the horizontal flips are taken as the second training set. During training, we first train a CNN with the first training set. We then use the first trained weight parameters to initialize the CNN and fine-tune them with the second training set.

The scale-related training method can improve the counting performance. Distinct from the image pyramid method, our proposed method trains CNNs in two stages according to the sample scale instead of training with multi-scale samples at the same time. The differences overcome the redundancy and patches overlap which can lead to overfit. We evaluate our training method on the ShanghaiTech Part_A and Part_B training sets with our SCNN. As shown in Table 1, our training method results in improvements in crowd counting accuracy and robustness.

5 Experiments

We evaluate our SCNN on three crowd counting benchmarks which belong to two datasets UCF_CC_50 [1] and ShanghaiTech [2]. The benchmarks are representative in terms of scale, scene and congestion level. Compared with the state-of-the-art methods, our model achieves competitive performance on accuracy and robustness. All the experiments are built on the CNN training framework named Caffe [37]. NVIDIA GTX TITAN X GPU cards are also used to accelerate the computation.

Table 1 Performance comparison of different training methods on ShanghaiTech dataset

Method	Part_A		Part_B	
	MAE	MSE	MAE	MSE
Training with the first training set	113.4	192.7	22.4	37.7
Training with the second training set	101.3	159.6	17.2	27.7
The scale-related training method	**90.8**	**134.1**	**16.8**	**27.4**

Bold entries indicate the best MAE/MSE performance of all methods

5.1 Evaluation metric

Following the common evaluation metrics [24], we choose the mean absolute error (MAE) and the mean squared error (MSE) to evaluate the accuracy and robustness of different methods. The MAE and MSE are formulated as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|z_i - \widehat{z_i}| \tag{7}$$

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

where N is the number of images in the test set. z_i and $\widehat{z_i}$ are the actual number of people and the estimated number of people in the i^{th} test image, respectively.

5.2 ShanghaiTech dataset part_A

ShanghaiTech dataset is the largest crowd counting dataset known to date, which contains 1198 annotated images for a total of 330,165 people. It is well-labeled by two-dimensional coordinates of the centers of people heads and consists of Part_A and Part_B. Part_A contains 482 images of high congestion level (Max 3139 people), which is collected from the Internet. There are 300 images in the Part_A training set and the remaining 182 images are employed to test.

The scale-related training method discussed in Section 4 is employed to augment the training set and train the network model. We choose the adaptive Gaussian kernel to generate the density maps, since the images in Part_A are high density. We use the SGD to optimize the network parameters with momentum of 0.9 and weight decay of 0.0005. The learning rate policy is set to step with the base learning rate of $1e^{-6}$. All the parameters are initialized by Gaussian weight initialization with a standard deviation of 0.01. Our method in action on an image of Part_A test set is shown in Fig. 6.

We compare our method with other 5 methods. Zhang et al. [24] first used a multi-task CNN to estimate the density map for crowd counting. A regression-based method is further compared with ours, which extracts Local Binary Pattern (LBP) features on the input images and predicts the number of people with ridge regression (RR). MCNN and MCNN based crowd count regression (MCNN-CCR)

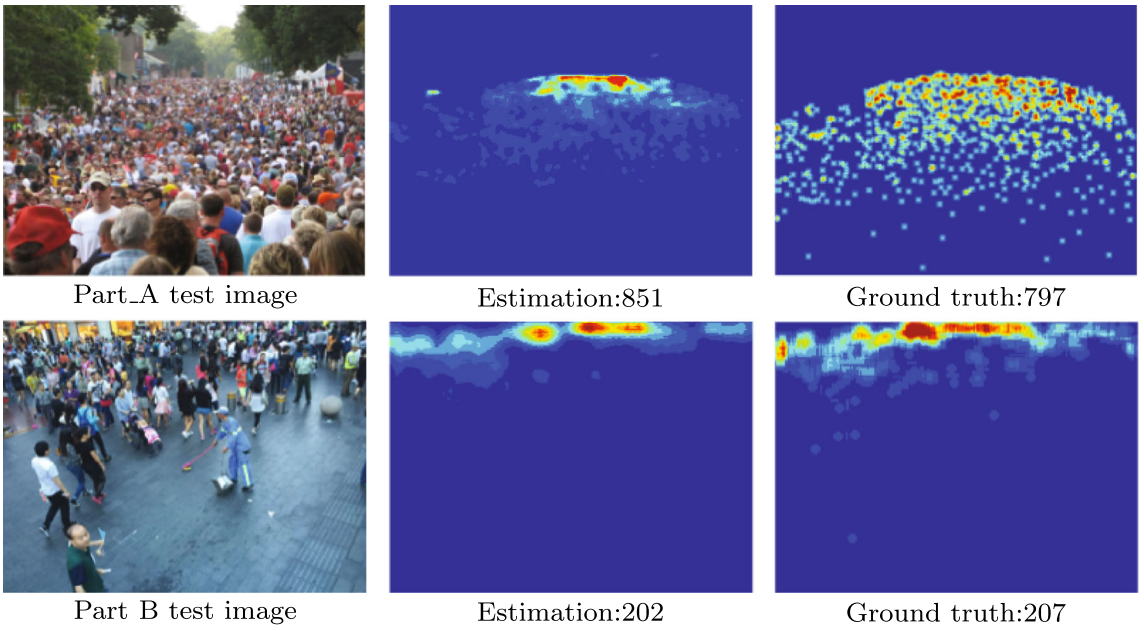

Fig. 6 The estimated density maps and ground truth density maps of our SCNN model on the ShanghaiTech test set

are proposed in [2]. We also compare our work with a single column FCN with a non-redundant training set augmentation scheme [36]. The performances of all the 5 methods on ShanghaiTech Part_A are provided in Table 2. The MAE and MSE of our method on Part_A are better than others.

5.3 ShanghaiTech dataset part_B

ShanghaiTech dataset Part_B is quite different from Part_A. Part_B contains 716 images, of which 400 images are for training and 316 images are for testing. The images of Part_B are medium congestion level (Max 578 people) and taken from the busy streets of Shanghai. Scale variations of these images are vast and it is very suitable to evaluate the performance of our method.

We also use the scale-related training method to augment the training set and train the model. All the hyper parameters for training and the network initialization parameters are the same as the Part_A. We choose the same spread in Gaussian kernel to generate ground truth density maps instead of the adaptive Gaussian kernel. The images in Part_B are relatively sparse and the distance of people in front of the camera is far away. Therefore the adaptive Gaussian kernel with one single β is difficult to describe the actual head sizes.

Table 2 Performances of different methods on ShanghaiTech dataset Part_A

Method	MAE	MSE
Zhang et al. [24]	181.8	277.7
LBP+RR	303.2	371.0
MCNN-CCR	245.0	336.1
MCNN	110.2	173.2
Marsden et al. [36]	126.5	173.5
Our method	**90.8**	**134.1**

Bold entries indicate the best MAE/MSE performance of all methods

Table 3 Performances of different methods on ShanghaiTech dataset Part_B

Method	MAE	MSE
Zhang et al. [24]	32.0	49.8
LBP+RR	59.1	81.7
MCNN-CCR	70.9	95.9
MCNN	26.4	41.3
Marsden et al. [36]	23.8	33.1
Our method	**16.8**	**27.4**

Bold entries indicate the best MAE/MSE performance of all methods

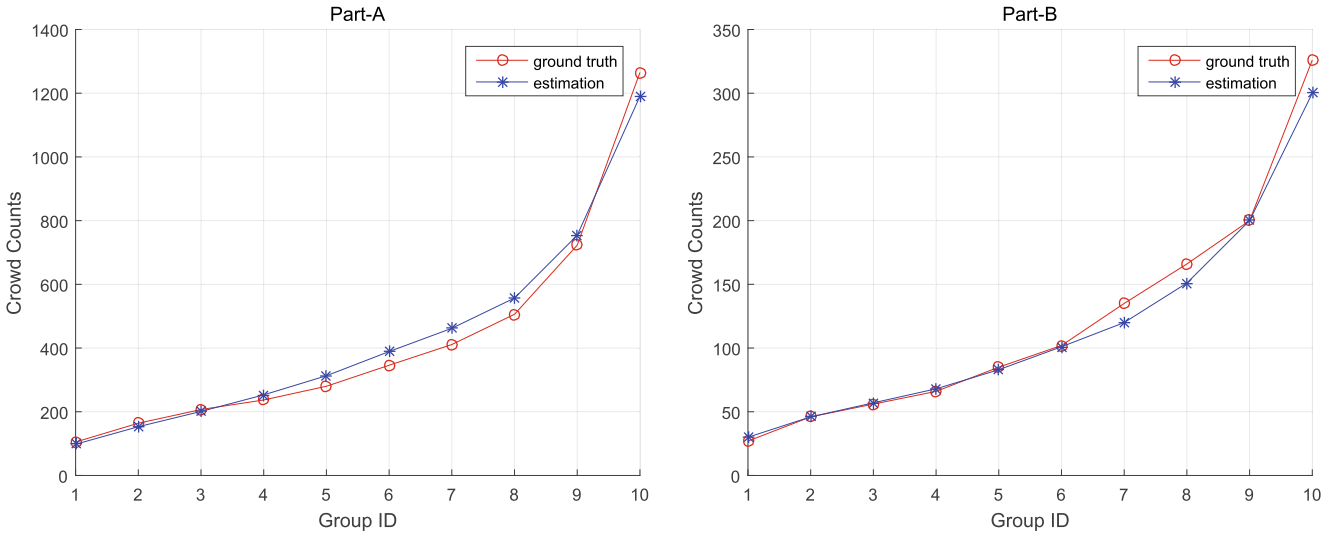

Fig. 7 Comparison of ground truth count and estimation count for images on ShanghaiTech dataset: We evenly divide test images into 10 groups according to increasing the number of people. Crowd count represents the average crowd number of images in each group

5 methods for evaluating Part_A are also used to evaluate Part_B. Table 3 shows the performance of 6 methods including ours on Part_B and our method achieves state-of-the-art performance. Figure 6 shows a test image in Part_B together with its estimated and ground truth density maps.

Following the work of [2], we also evenly divide the test images in Part_A and Part_B into 10 groups in ascending order of the number of people, in order to further analyze the performance of our method. We compare the actual count and estimation count of each group and plot a line chart as shown in Fig. 7. The fold lines of actual count and estimation count are very close. The result demonstrates that our method is of high accuracy and robust to large variation. In the medium congestion level crowd scene, the estimation count of our method is almost the same as the actual count. It proves that our method is effective in overcoming scale variations.

5.4 UCF_CC_50 dataset

The UCF_CC_50 dataset is a very challenging crowd counting dataset due to high crowd density and less training images. It only contains 50 images collected from the Internet and the crowd in the image is extremely congested. The crowd count of each image is between 94 and 4543 with an average of 1280 people per image. As the dataset publishers have taken, we perform a 5-fold cross validation. To augment the training set, we crop 4 patches from each training image without overlap and each patch is 1/4 size of the image. The adaptive Gaussian kernel is also used.

We compare our method with other 8 methods on the UCF_CC_50 dataset. The first three methods [1, 15, 38] adopted the handcraft features and regression models to estimate the people count of the input images. The last five [2, 4, 24, 25, 39] are CNN-based methods with the multi-scale network architecture. Table 4 illustrates that our method achieves the best MAE and competitive MSE. The experimental results prove that our method is still effective for high crowd density. Figure 8 shows a test image in UCF CC 50 together with its estimated and ground truth density maps.

Table 4 Performances of different methods on UCF_CC_50 dataset

Method	MAE	MSE
Rodriguez et al. [38]	655.7	697.8
Lempitsky et al. [15]	493.4	487.1
Idrees et al. [1]	419.5	541.6
Zhang et al. [24]	467.0	498.5
CrowdNet [4]	452.5	–
Hu et al. [25]	431.5	**438.5**
MCNN [2]	377.6	509.1
Zeng et al. [39]	363.7	468.4
Our method	**346.6**	477.5

Bold entries indicate the best MAE/MSE performance of all methods

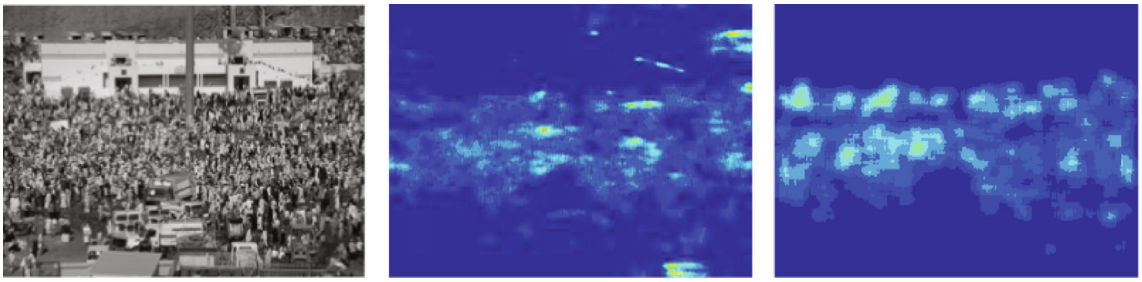

Fig. 8 The estimated density map and the ground truth density map of our SCNN model on the UCF_CC_50 test set

6 Conclusion

In this paper, we proposed a Skip-connection Convolutional Neural Network for still image crowd counting, which can achieve high counting accuracy and robustness facing the issues of scale variations and perspective. We evaluate our model on three crowd counting benchmarks and our method outperforms the state-of-the-art crowd counting methods. A scale-related training method is also proposed suppressing the influence of scale variations. The experimental results demonstrate the effectiveness of our method. In the future work, we will further study the CNN-based crowd counting in the surveillance video which is close to industrial applications.

Acknowledgments This work is supported in part by the National Natural Science Foundation of China under grant No. 61233003, in part by the Equipment Pre-research Fund under grant No. 61403120201.

References

1. Idrees H, Saleemi I, Seibert C, Shah M (2013) Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2547–2554
2. Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 589–597
3. 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, vol 1, p 6
4. 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, pp 640–644. ACM
5. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3431–3440
6. Onoro-Rubio D, López-Sastre RJ (2016) Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision, pp 615–629. Springer
7. Lin S-F, Chen J-Y, Chao H-X (2001) Estimation of number of people in crowded scenes using perspective transformation. IEEE Trans Syst Man Cybern Syst Hum 31(6):645–654
8. Wu B, Nevatia R (2005) Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: 2005 10th IEEE International Conference on Computer Vision, 2005. ICCV, vol 1, pp 90–97. IEEE
9. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR, vol 1, pp 886–893. IEEE
10. Wang M, Wang X (2011) Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3401–3408. IEEE
11. Ge W, Collins RT (2009) Marked point processes for crowd counting. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR, pp 2913–2920. IEEE
12. Li M, Zhang Z, Huang K, Tan T (2008) Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 2008 ICPR 2008, 19th International Conference on Pattern Recognition, pp 1–4. IEEE
13. Chan AB, Liang Z-SJ, Vasconcelos N (2008) Privacy preserving crowd monitoring: Counting people without people models or tracking. In: CVPR 2008. IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp 1–7. IEEE
14. Chen K, Loy CC, Gong S, Xiang T (2012) Feature mining for localised crowd counting. In: fBMVC, vol 1, p 3
15. Lempitsky V, Zisserman A (2010) Learning to count objects in images. In: Advances in Neural Information Processing Systems, pp 1324–1332
16. Chan AB, Vasconcelos N (2009) Bayesian poisson regression for crowd counting. In: 2009 IEEE 12th International Conference on Computer Vision, pp 545–551. IEEE
17. Kong D, Gray D, Tao H (2006) A viewpoint invariant approach for crowd counting. In: ICPR 2006. 18th International Conference on Pattern Recognition, 2006, vol 3, pp 1187–1190. IEEE
18. 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. IEEE
19. Chan AB, Vasconcelos N (2012) Counting people with low-level features and bayesian regression. IEEE Trans Image Process 21(4):2160–2177
20. Paragios N, Ramesh V (2001) A mrf-based approach for real-time subway monitoring. In: 2001 IEEE Computer Society Conference

on Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the, vol 1, pp I–I. IEEE
21. Regazzoni CS, Tesei A (1996) Distributed data fusion for real-time crowding estimation. Signal Process 53(1):47–63
22. Bell S, Lawrence Zitnick C, Bala K, Girshick R (2016) Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2874–2883
23. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
24. Zhang C, Li H, Wang X, Yang X (2015) Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 833–841
25. Hu Y, Chang H, Nian F, Wang Y, Li T (2016) Dense crowd counting from still images with convolutional neural networks. J Vis Commun Image Represent 38:530–539
26. Zhang Y, Chang F, Wang M, Zhang F, Han C (2017) Auxiliary learning for crowd counting via count-net. Neurocomputing
27. Sindagi VA, Patel VM (2017) Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. arXiv:1707.09605
28. Chen L-C, Yang Y, Wang J, Xu W, Yuille AL (2016) Attention to scale: Scale-aware semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3640–3649
29. Neverova N, Wolf C, Taylor GW, Nebout F (2014) Multi-scale deep learning for gesture detection and localization. In: Workshop at the European Conference on Computer Vision, pp 474–490. Springer
30. Eigen D, Puhrsch C, Fergus R (2014) Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems, pp 2366–2374
31. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision, pp 2650–2658
32. Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35(8):1915–1929
33. Zeiler MD, Ranzato M, Monga R, Mao M, Yang K, Le QV, Nguyen P, Senior A, Vanhoucke V, Dean J et al (2013) On rectified linear units for speech processing. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 3517–3521. IEEE
34. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2818–2826
35. Dumoulin V, Visin F (2016). arXiv:1603.07285
36. Marsden M, McGuiness K, Little S, O'Connor NE (2016) Fully convolutional crowd counting on highly congested scenes. arXiv:1612.00220
37. 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: Proceedings of the 22nd ACM international conference on Multimedia, pp 675–678. ACM
38. Rodriguez M, Laptev I, Sivic J, Audibert J-Y (2011) Density-aware person detection and tracking in crowds. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp 2423–2430. IEEE
39. Zeng L, Xu X, Cai B, Qiu S, Zhang T (2017) Multi-scale convolutional neural networks for crowd counting. arXiv:1702.02359

Luyang Wang received the B.S. degree in automation from the University of Science and Technology of China, Hefei, China, in 2015. He is currently pursuing the Ph.D. degree in the Department of Automation, University of Science and Technology of China. His recent research interests include crowd counting and neural network compression.

Baoqun Yin received the B.S. degree in fundamental mathematics from Sichuan University, Chengdu, China, in 1985, his M.S. degree in applied mathematics from University of Science and Technology of China, Hefei, China, in 1993, and his Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 1998. He is currently a Professor in the Department of Automation, University of Science and Technology of China, Hefei, China. His current research interests include discrete event dynamic systems, deep learning, and network resource management and optimization.

Aixin Guo received the B.S. degree in automation from the Anhui University, Hefei, China, in 2015. She is currently pursuing her master degree in the Department of Automation, University of Science and Technology of China. Her research interest is pedestrian recognition.

Hao Ma received the B.S. degree in automation from the University of Science and Technology of China, Hefei, China, in 2015, where he is currently working toward the master's degree with the Department of Automation, University of Science and Technology of China. He focuses on computer vision and pattern recognition.

Jie Cao received the B.S. degree in measuring and control technology and instrumentations from the Anhui University, Hefei, China, in 2012, and his Ph.D. degree in the Department of Automation from University of Science and Technology of China, Hefei, China, in 2017. His research interests include Bayesian networks and swarm intelligence.