Improving crowd counting with scale-aware convolutional neural network

Qingge Ji\textsuperscript{1,2} | Hang Chen\textsuperscript{1,2} | Di Bao\textsuperscript{1,2}

\textsuperscript{1} School of Data and computer Science, Sun Yat-sen University, Guangzhou, China
\textsuperscript{2} Guangdong Province Key Laboratory of Big Data Analysis and Processing, Guangzhou, China

Correspondence
Qingge Ji, School of Data and computer Science, Sun Yat-sen University, Guangzhou, 510006, China. Email: issjqg@mail.sysu.edu.cn

Funding information
Natural Science Foundation of Guangdong Province, Grant/Award Number: No. 2016A03013288

Abstract
Large-scale variations may cause a serious problem in crowd counting. In recent years, most methods for this problem use convolutional neural networks with a fixed scale for encoding and decoding image features. The scale of the convolutional layer is usually manually adjusted and may have to deal with image features on unfitted scales. In this paper, a method called scale-aware convolutional neural network (SCNet) is proposed, which adds a scale selection mechanism to the dilated convolutional operation. Shared weight multi-branch is used to deal with features on different scales, and an attention mechanism is introduced to determine the weights of the branches that fit the scale. Experimental results demonstrate that the proposed SCNet outperforms most existing methods.

1 | INTRODUCTION

In public places, such as plazas, supermarkets, and subway stations, there may be cases of excessive crowds. Dense crowds may result in accidents such as stampede. For places where public accidents may occur, an automatic monitoring system of crowd in these scenarios is required from the perspective of urban management. Crowd counting in public places has gradually gained people’s attention in recent years, which plays an important role in disaster prevention, public placement design, intelligent personnel scheduling and other aspects.

In recent years, convolutional neural networks (CNN) have achieved significant improvements for crowd counting problem. Crowd counting methods based on density map \cite{1,3} are widely used by researchers in the field of crowd counting and yield better results than regression \cite{4} and detection \cite{5} based methods.

Crowd counting methods usually suffer from scene diversity, occlusion, and the scale variation of pedestrian, especially in large scale range. The heads’ sizes of nearby and distant pedestrians in some images also vary greatly, increasing the dimensions of crowd counting. The existing CNN models rarely have the ability to deal with such a large-scale-range problem.

The methods in \cite{1,3} use the dynamic sigma of Gaussian filter to generate density map to mitigate the performance penalty of perspective effects. The sigma is assigned according to the distance between people, and larger distance is corresponded to larger Gaussian filter sigma. However dynamic sigma of Gaussian filter increases the dimensions of target space. For different crowds, the model needs to generate density maps of multi-scale circular areas.

In the target space of fixed sigma, the scale of the activation is fixed. When we apply the dynamic sigma, the scale of activation will be various accordingly. Thus the task to transform from source space to target space becomes even harder. On the other hand, when the fixed Gaussian sigma is used \cite{2}, the scale of pedestrian in the source space and density map in the target space will become mismatched, affecting the performance of the model.

To deal with the large-scale-range problem, we propose a scale-aware CNN, called SCNet for short. It increases the number of branches without raising the parameters in the meantime. In order to make the convolutional layers adaptive to images’ features on different scales, different dilated rates are used on each branch. The dilated rate of the original convolutional kernel is 1, but we set dilated rate to be 0, 1 and 2 respectively in
SCNet. In addition, each branch can produce an output with an inappropriate scale. We thus add an attention module to make a selection. The attention modules select an output of the branch with a suitable scale by giving different branches a unique importance weight. The contributions in this paper are two folds:

1. We propose an effective structure, SCNet, to deal with large-scale-range crowd counting problem. SCNet only adds a few parameters because multiple branches share weights in CNN, which can be fine-tuned from the original convolutional layers.

2. The proposed SCNet outperforms the state-of-the-art methods on two public datasets, ShanghaiTech [1] and UCF-QNRF [6]. We got 62.4 MAE on ShanghaiTech Part A, and 8.2 MAE on ShanghaiTech Part B. For UCF-QNRF dataset, we attained 103.9 MAE, a large improvement over the previous results [2, 3].

2 RELATED WORK

2.1 Multi-scale method

To alleviate large-scale problem of crowd counting, early methods use image pyramid. Hydra-CNN [7] uses a pyramid structure of input patches to do a perspective-free object counting, but it relies on long-time training and inference stage.

Some existing methods use multi-column structure which adopts different kernel sizes in different columns. MCNN [1] uses three columns consisting of little, medium and large perception fields, and fuses the feature maps through $1 \times 1$ convolutional layers to generate the density map at the end. CrowdNet [8] combines the shallow and deep CNN to extract the high-level semantic message and the low-level pattern of head patch in parallel. Switch-CNN [9] uses a model to classify the density level of the image and then decides which branch to use. Based on MCNN [1], CPCNN [10] adds two extra columns to incorporate global context and local context so as to generate high-quality density map with contextual pyramid.

The other methods aggregate the multi-column into a single column. MSCNN [11] and SANet [2] combine CNNs with different size of kernels into a block. The feature maps are merged in every block similar to Inception module [12]. Inspired by the ideas of these methods, the convolutional layers of our proposed SCNet has multiple dilate rates and are fused into a block together.

2.2 Attention mechanism

Attention mechanism has been used in image classification [13–16] especially fine-grain image classification [17, 18], semantic segmentation [19, 20], object detection [21, 22], and re-identification [23, 24]. In crowd counting, DecideNet [25] uses attention mechanism to decide the weight between density maps generated by regression network and detection network. FCN-7c-3s [26] generates an across-scale attention to fuse the density maps at each scale. In our work, the attention mechanism is used to make a selection among the output of branches with different receptive fields.

2.3 Dilated convolution

Dilated convolution is widely used in semantic segmentation [27, 28]. It can enlarge the receptive fields of the model without adding any parameters. In crowd counting, CSRNet [3] uses the dilated convolution to get the larger receptive fields and to replace pooling operations. AMDCN [29] is an Aggregated multicolumn dilated convolutional network that can increase the ability of the network to selectively aggregate multi-scale information. In our model, we combine weight sharing convolutional layers with different dilated rates to capture image features at different scales.

2.4 Other crowd counting methods

Comparing with other fields of computer vision such as image classification and semantic segmentation, crowd counting usually lacks data with labels, so some unsupervised methods have to be used to improve the performance. GWTA-CCNN [30] uses the auto-encoder to learn the reconstruction of images. Most of its parameter can be trained without any labelled data. L2R [31] sets up a self-supervised task to make a comparison of counting number between an image and its patches, which adds constraints based on prior knowledge that the number of crowd in the whole image must be greater than that in the local patches.

Refinement strategy is used in some crowd counting methods. After predicting the density map, CNN-Boosting [32] adds network to learn the residual error of earlier density maps in the boosting process. IC-CNN [33] uses low-resolution and high-resolution CNN branches to predict the density map iteratively. It first generates low-resolution density map by LR-CNN. Then the density map is passed to the HR-CNN to get the high-resolution density map. DRSAN [34] uses STN [35] to make spatial transformation on density map to simulate variance in scale of rotation before refinement. These methods can generate high quality density map but suffer from high cost of inference time.

Some crowd counting methods combine multiple tasks. In CMTL [36], the two tasks, density map regression and counting number regression, are performed in the meantime. DecideNet [25] adds detection mechanism to the density map regression task and fuses the output of two tasks according to the attention map. Composition-loss [6] combines three tasks consisting of counting, density map prediction and crowd localisation.

Other crowd counting methods focus on ensemble learning. Based on one multi-branch frames, DConvNet [37] adds constraints to increase the negative correlation of each branch’s output. IG-CNN [38] clusters the density levels of images, and then
builds multiple network structures to predict images with different density levels.

3 | METHODOLOGY

Our proposed method uses a CNN to extract features of crowd images on multiple receptive fields, while keeping limited parameters to remain a great performance. The method’s description is divided into two parts: structure and module.

3.1 | Framework of SCNet

In this part we introduce our proposed SCNet in Figure 1, in which we modify the origin dilated convolutional layers of our baseline, CSRNet [3] (denoted as CSRNet* in tables). We also used first 10 convolutional layers of VGG-16 [4] as front end in SCNet, as shown in Table 1. In order to enable the other convolutional layers to process image features on different scales, we use the shared weight multi-branch structure to deal with the input feature and the attention module to select and merge the output of the multi-scale columns. We design a scale-aware convolutional module(SCM) consisting of a multi-branch layer and an attention module, as shown in Figure 2. In the end, six SCMs are cascaded as the back end of SCNet. We will describe the details in the following SCM.

3.2 | Scale-aware convolutional module

The SCM consists of two parts: shared weight multi-branch structure and attention module. Dilated convolutional kernel is applied in multi-branch to expand the receptive field in each branch and the attention mechanism is to adjust the weights of multiple branches.

3.2.1 | Shared weight multi-branch

We use \( C \) branches with \( k \times k \) kernel size in SCNet, which play the roles of back end in our model. In [39], there are two branches in the Siamese network that share exactly the same architecture and the set of weights. Inspired by this model, we make each branch share the same weight. For normal convolutional kernel, a kernel’s number of parameters is \( C \times (k \times k + 1) \). However, if we use weight sharing mechanism, \( C \) branches use the same parameters for forward and back propagation, where the parameters’ number is \( k \times k + 1 \), like a single column network.

Weight sharing mechanism has benefits as follows: (1) fewer parameters; (2) better performance; and (3) replaceable convolutional layer. Shared weight can be regarded as a scalable convolutional kernel, which convolves the sampling centre with different radius on different scales. The size relationship among the outputs is beneficial to the model for obtaining the texture or structure information of images. The model can be trained first.
based on normal convolutional layer, and then fine-tuned after replaced by shared weight multi-branch.

3.2.2 Dilated convolutional layer

The $C$ branches have different dilated rate $d_i$, where $i$ represents the index of branch.

Our baseline uses a single branch, with $3 \times 3$ kernel size and dilated rate $= 2$. However, SCM uses three branches of different kernel sizes, following the kernel size with our baseline. In our experiment, we set $C = 3$, and the dilated rates are set as $d_0 = 0, d_1 = 1, d_2 = 2$, respectively, as shown in Table 1. The receptive field of normal convolutional layer with $3 \times 3$ kernel size is also $3 \times 3$. But for convolutional kernel which has dilated rate $= 2$ with the same kernel size, the receptive field is $5 \times 5$. To get the smaller receptive field, we design the convolutional kernel with dilated rate $= 0$, which means that its receptive field is $1 \times 1$.

Since the dilated rate $0$ is unusual thus not in [3], we will introduce it in detail next. For a normal $3 \times 3$ convolutional kernel on the feature map at the location of $(x, y)$, we have

$$O_{(x,y)} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I_{(x+i,y+j)} \ast w_{(i+1,j+1)} + b$$  

(1)

where $O_{(x,y)}$ represents the output feature at the location $(x, y)$, and $I_{(x+i,y+j)}$ represents the input feature at the location $(x+i, y+j)$. The symbols $w$ and $b$ represent the weights and bias of the $3 \times 3$ convolutional kernel, respectively.

As for $3 \times 3$ convolutional kernel with dilated rate $d$, we have

$$O_{(x,y)} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I_{(x+i,d,y+j,d)} \ast w_{(i+1,j+1)} + b$$  

(2)

When $d = 1$, Equations (1) and (2) are equivalent. Now let $d = 0$, we can get

$$O_{(x,y)} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I_{(x+i,y+j)} \ast w_{(i+1,j+1)} + b$$

$$= \sum_{i=-1}^{1} \sum_{j=-1}^{1} I_{(x,y)} \ast w_{(i+1,j+1)} + b$$

$$= \sum_{i=-1}^{1} \sum_{j=-1}^{1} (I_{(x,y)} \ast w_{(i+1,j+1)} + \frac{b}{g})$$

$$= \sum_{k=1}^{9} (I_{(x,y)} \ast w'_{k} + b'_{k})$$

where $w'_{k}$ and $b'_{k}$ represent the weight and bias of $1 \times 1$ convolutional kernel. The last function shows that one $3 \times 3$ convolutional kernel with dilated rate $0$ is equivalent with nine $1 \times 1$ convolutional kernels. Thus for the branch with dilated rate $= 0$, we use nine $1 \times 1$ convolutional kernels to share weight with other two branches.

3.2.3 Attention module

An attention module is used to be the gate of a SCM. The module contains a $3 \times 3 \times C$ convolutional layer and a softmax layer, which can be regarded as a 2-layer multi-branch subnet-work, as shown in Figure 3. The input of the attention module is the output of previous layer of SCNet, and the output of first layer is a $C$ channel’s attention map. Then we perform a softmax operation on the dimension of channel to make sure the summation of different channel output values on the

![FIGURE 2: Structure of SCM. The convolutional layer contains three types convolutional kernels whose dilated rate are 2, 1 and 0, respectively. And the attention module generates three attention maps to re-weight the outputs from convolutional layer and concatenate them into a feature map.](image1.png)

![FIGURE 3: Structure of attention module. The first layer is a convolutional layer containing $C$ filters with three kernel size if there are $C$ branches in the backbone. Then though a softmax layer to make sure the summation of value is one. The final outputs of attention module are $C$-channel feature maps as attention maps. Here we set $C = 3$.](image2.png)
same location of attention maps becomes 1. Softmax operation process is as Equation (4)

\[
\text{softmax}(I)_{i,j,c} = \frac{e^{I_{i,j,c}}}{\sum_{c=1}^{C} e^{I_{i,j,c}}}
\]  

(4)

where \( I \) represents the input of softmax layer, and \( i, j, c \) represent the coordinate of row, column and channel in feature maps, respectively. \( C \) represents the total number of channels and in this paper, \( C = 3 \). Since the attention maps’ value is normalized into \([0,1]\), it can assign weights to each branch by element-wise multiplication at different scales.

The role of attention module includes two aspects. On one hand, compared with manually setting dilated rate in CSRNet, it enables the model to automatically select the dilated rate that is most suitable for the dataset. On the other hand, for a single image with a large scale range, it can give a scale-aware selection for each pixel.

4 | EXPERIMENTS

In this section we will first introduce four datasets and two evaluation metrics in crowd counting. Then we will evaluate the proposed approach on these four datasets with two metrics. More extra experiments are conducted to confirm the effect of mentioned modules.

4.1 | Evaluation metrics

Mean absolute error (MAE) and mean squared error (MSE) are mainly used to evaluate the approaches. They are defined as follows:

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\text{cnt}^e_{ij} - \text{cnt}^g_{ij}|
\]

(5)

\[
\text{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} |\text{cnt}^e_{ij} - \text{cnt}^g_{ij}|^2}
\]

(6)

where \( N \) represents the total number of images, \( i \) represents the index of the image, \( \text{cnt}^e_{ij} \) and \( \text{cnt}^g_{ij} \) represent the estimated head count and the ground truth of the image respectively.

4.2 | Training details

4.2.1 | Ground truth generation

The ground truth is generated by blurring the head annotations with a normalized Gaussian kernel (sum to one) following [1].

\[
F(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_\sigma(x)
\]

(7)

where \( \delta(x - x_i) \) is 1 on the locations of people’s head, and is 0 on other locations. Then \( \delta(x - x_i) \) is convolved with a Gaussian kernel \( G_\sigma(x) \). We use a fixed Gaussian kernel \( \sigma \) to each dataset. For ShanghaiTech Part B we use \( \sigma = 4 \), and for other datasets we use \( \sigma = 3 \).

4.2.2 | Data augmentation

For each image, we first crops 9 patches, where the size of patches is 1/4 of the original image. Among the 9 patches, 4 are from the four corners of the original image, and the other 5 are from random locations of the image. After cropping out 9 patches, we flip them horizontally, which doubles the number of training sets.

4.2.3 | Optimizer

We use SGD with a learning rate 1e-7 in our training following the setting of CSRNet.

4.2.4 | Loss function

We use Euclidean loss in our training. The loss is defined as follow:

\[
L_{ij} = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} ||M^e_{ij} - M^g_{ij}||^2
\]

(8)

where \( i \) and \( j \) represent and index of rows and columns in an image, \( M^e_{ij} \) and \( M^g_{ij} \) represent the value of one pixel of estimated density map and ground truth density map.

4.2.5 | Experimental environment

The GPU used in the experiment is NVIDIA GTX 1080Ti and the operating system is Ubuntu. The implementation of SCNet is based on the PyTorch framework.

4.3 | Comparison with the state-of-the-art methods

4.3.1 | ShanghaiTech dataset

The ShanghaiTech dataset [1] has 1198 images with 330,165 people totally. It contains two parts, Part A and Part B. Part A dataset has 482 images crawled from the Internet, in which 300 images are used for training and 182 images for testing. Part B dataset has 716 images from busy streets of metropolitan areas in Shanghai, where 400 images are used for training and 300 images for testing. As shown in Table 2, our method gets 62.4 MAE on Part A, which is 4.6 MAE less than SANet.
| Method       | S1  | S2  | S3  | S4  | S5  | Average |
|-------------|-----|-----|-----|-----|-----|---------|
| MCNN [1]    | 3.4 | 20.6| 12.9| 13.0| 8.1 | 11.6    |
| Switch-CNN  [9] | 4.4 | 15.7| 10.0| 11.0| 5.9 | 9.4     |
| CSRNet [3]  | 2.9 | 11.5| 8.6 | 16.6| 3.4 | 8.6     |
| SANet [2]   | 2.6 | 13.2| 9.0 | 13.3| 3.0 | 8.2     |
| DRSAN [34]  | 2.6 | 11.8| 10.3| 10.4| 3.7 | 7.76    |
| CSRNet*     | 2.08| 14.1| 12.7| 9.37| 3.37| 8.24    |
| SCNet(Ours) | 1.7 | 13.4| 10.7| 18.7| 3.1 | 9.5     |

There is also a sample from Part A shown in Figure 4. From top to bottom row are the images, density maps generated by CSRNet* and SCNet from left to right.

**FIGURE 4** A sample from ShanghaiTech Part A. The second and third rows are the density maps and heat maps generated by CSRNet* and SCNet.
TABLE 5  Results on UCSD dataset

| Method       | MAE  | MSE  |
|--------------|------|------|
| MCNN [1]     | 1.07 | 1.35 |
| Switch-CNN [9] | 1.62 | 2.10 |
| CSRNet [3]   | 1.16 | 1.47 |
| SANet [2]    | 1.02 | 1.29 |
| CSRNet*      | 1.38 | 2.32 |
| SCNet(Ours)  | 1.12 | 1.74 |

TABLE 6  Ablation study on ShanghaiTech dataset (Param's unit is in millions)

| Method               | Part A    | Part B     | Params |
|----------------------|-----------|------------|--------|
|                      | MAE   | MSE   | MAE   | MSE   |        |
| CSRNet*              | 68.6  | 106.1 | 10.0  | 15.9  | 16.26  |
| W/o share+add        | 69.6  | 111.7 | 8.5   | 13.5  | 33.51  |
| W/o share+attention  | 68.1  | 109.5 | 8.5   | 13.4  | 33.58  |
| Share+add            | 65.2  | 100.0 | 8.4   | 12.7  | 16.26  |
| Share+attention(ours)| 62.4  | 101.7 | 8.2   | 13.0  | 16.32  |

4.3.4  UCSD dataset

The UCSD dataset [44] is a sparse density dataset that is a 2000-frame video dataset chosen from one surveillance camera on the UCSD campus. The ROI and the perspective map are provided in the dataset. The resolution of each image is $238 \times 158$, and the crowd count in each image varies from 11 to 46. Following Chan et al. did, we also use frames from 601 to 1400 as the training set and the remaining frames for testing. We show the accuracy in Table 5, where it only reduces MAE 0.04 by CSRNet’s, but does not acquire the best result in comparing list. Through observing the difference from previous datasets, UCSD dataset is grayscale, where the deficiency of colours might influence the distribution of attention module and give multiple branches a mediocre selection.

4.4  Ablation study

We have also conducted some experiments to demonstrate the effect of weight sharing and attention module in the proposed results on ShanghaiTech and UCF-QNRF dataset. To compare the effect of attention module, we keep multi-branch structure and simply add the output of three branches together. And to compare the effect of weight sharing, we make three branches to have their own weights. 

The results of ablation study are shown in Tables 6 and 7. We recombine the structures that have attention module(attention), simply adding its multi-branch at the end(add), and use weight sharing mechanism(share). The structure of simple addition without weight sharing(w/o share+add) performs worse than structure with weight sharing(share+add) or with weight sharing and attention module(w/o share+attention).

Weight sharing and attention module both play important roles in improving performance of the model. When using the weight sharing based on simply adding strategy, MAE on ShanghaiTech A decreases from 69.6 to 65.2, and for ShanghaiTech B MAE decreases from 8.5 to 8.4. On UCF-QNRF, the MAE descends from 111.5 to 109.6. When using the weight sharing strategy based on attention module, the MAE drops from 68.1 to 62.4 on ShanghaiTech A and from 8.5 to 8.2 on ShanghaiTech B. As for UCF-QNRF dataset, the MAE decreases to 103.9, which is 5.7 less than the MAE 109.6.

We also show the number of parameters in Table 6, where our model can reduce the number of parameters greatly comparing with models without weight sharing, but has some extra parameters relative to CSRNet because of attention modules. It turns out that the effect of weight sharing can reduce the number of parameters. The same parameters make the output of different branches comparable, which is more conducive to model to learn more useful attention maps.

When attention mechanism is used without weight sharing, on ShanghaiTech A and UCF-QNRF, the MAE reduces by 1.5 and 1.8, respectively. While on ShanghaiTech B, attention module keeps its performance. When it is used with weight sharing, the model gets 2.7 MAE reduction on ShanghaiTech A, 0.2 MAE reduction on ShanghaiTech B and 5.7 MAE reduction on UCF-QNRF. This result indicates that the attention mechanism relies on weight sharing strategy to make a more scale-aware selection.

4.5  Analysis of attention map

Figure 5 visualizes the activation blocks of attention maps in each SCM. Some blocks act like dilated rate selectors, which can find more suitable dilated rate and branches’ weights for their dataset. In these blocks, the activation value is bias towards certain specify dilated rate, and the other blocks generate spatial attention maps. In ShanghaiTech Part A and UCF-QNRF, the 5th block generates the spatial attention maps. And in ShanghaiTech Part_B, the 4th block generates the spatial attention maps.

We calculate the summation of attention maps’ activation value and list them in Tables 8, 9 and 10 respectively, which can be seen that the distribution of activation value among branches is very uneven. For example, in Table 8, the branch with dilated
FIGURE 5  Attention maps of the multi-branch. Three images are from ShanghaiTech Part A, ShanghaiTech Part B and UCF_QNRF dataset. An attention map block has six rows and three columns. Each row represents the activation value of one block in our SCM. From the first row to the last row corresponds to the block from the closer to the input of model to the closer to the output of model. Three columns correspond to dilated rate 0, 1 and 2, respectively.

TABLE 8  Activation value of attention maps in ShanghaiTech Part A dataset. Column 0, 1 and 2 refer to the dilated rate of its corresponding branches.

| Layer | 0    | 1    | 2    |
|-------|------|------|------|
| 1st   | 0.11 | 0.14 | 0.76 |
| 2nd   | 0.01 | 0.01 | 0.98 |
| 3rd   | 0.97 | 0.00 | 0.02 |
| 4th   | 0.01 | 0.06 | 0.93 |
| 5th   | 0.60 | 0.40 | 0.01 |
| 6th   | 0.95 | 0.01 | 0.04 |

TABLE 9  Activation value of attention maps in ShanghaiTech Part B dataset.

| Layer | 0    | 1    | 2    |
|-------|------|------|------|
| 1st   | 0.60 | 0.40 | 0.00 |
| 2nd   | 1.00 | 0.00 | 0.00 |
| 3rd   | 0.00 | 0.99 | 0.01 |
| 4th   | 0.48 | 0.01 | 0.51 |
| 5th   | 0.00 | 0.49 | 0.51 |
| 6th   | 0.06 | 0.03 | 0.91 |

TABLE 10  Activation value of attention maps in UCF_QNRF dataset.

| Layer | 0    | 1    | 2    |
|-------|------|------|------|
| 1st   | 0.32 | 0.58 | 0.10 |
| 2nd   | 0.02 | 0.02 | 0.96 |
| 3rd   | 0.02 | 0.00 | 0.98 |
| 4th   | 0.12 | 0.86 | 0.02 |
| 5th   | 0.01 | 0.98 | 0.01 |
| 6th   | 0.63 | 0.01 | 0.36 |

rate 2 in the 2nd layer gets the dominant attention activation value, so does the branch with dilated rate 2 in the 3rd layer in the sample of Table 10. A layer with a very uneven branch activation distribution indicates that the role played by this layer is mainly branch selection, while the block of other layers plays a role of spatial selection.

5  | CONCLUSION

In this paper, we have proposed a novel model called SCNet for crowd counting. In the proposed model we use multi-branch
dilated convolutional layers with shared weights for feature processing, and an attention module for branch selection and space selection. By analysing the information learned from attention module, the hyper-parameter settings of CSRNet are modified using the automatically learned dilated rate. Experiments are conducted on the public crowd counting datasets, where the performance of the redesigned model exceeds CSRNet on ShanghaiTech Part A and Part B and achieves significant improvement compared with most existing methods.

ACKNOWLEDGMENTS
This research was supported by the Natural Science Foundation of Guangdong Province, China (No. 2016A030313288).

REFERENCES
1. Zhang, Y., et al.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 589–597.IEEE, Piscataway, NJ (2016)
2. Cao, X., et al.: Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European Conference on Computer Vision, pp. 734–750.Springer, Berlin (2018)
3. Li, Y., Zhang, X., Chen, D.: Crsnet: Dilated convolutional networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1100.IEEE, Piscataway, NJ (2018)
4. Idrees, H., et al.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547–2554.IEEE, Piscataway, NJ (2013)
5. Li, M., et al.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 2008 19th International Conference on Pattern Recognition, pp. 1–4.IEEE, Piscataway, NJ (2009)
6. Idrees, H., et al.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 532–546.Springer, Berlin (2018)
7. Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision, pp. 615–629.Springer, Berlin (2016)
8. Boominathan, L., Kruhviintsi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 24th ACM international conference on Multimedia, pp. 640–644.ACM Press, New York, NY (2016)
9. Sam, D.B., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4031–4039.IEEE, Piscataway, NJ (2017)
10. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid cnns. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1861–1870.IEEE, Piscataway, NJ (2017)
11. Zeng, L., et al.: Multi-scale convolutional neural networks for crowd counting. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 465–469.IEEE, Piscataway, NJ (2017)
12. Szegedy, C., et al.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.IEEE, Piscataway, NJ (2016)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.MIT Press, Cambridge, MA (2012)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:14091556 (2014)
15. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.IEEE, Piscataway, NJ (2015)
16. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.IEEE, Piscataway, NJ (2016)
17. Fu, J., Zheng, H., Mei, T.: Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4438–4446.IEEE, Piscataway, NJ (2017)
18. Zheng, H., et al.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5209–5217.IEEE, Piscataway, NJ (2017)
19. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241.Springer, Berlin (2015)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.IEEE, Piscataway, NJ (2015)
21. Ren, S., et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99.MIT Press, Cambridge, MA (2015)
22. Redmon, J., et al.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.IEEE, Piscataway, NJ (2016)
23. Li, W., et al.: Deepreid: Deep filter pairing neural network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 152–159.IEEE, Piscataway, NJ (2014)
24. Ahmed, F., Jones, M., Marks, T.K.: An improved deep learning architecture for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3908–3916.IEEE, Piscataway, NJ (2015)
25. Liu, J., et al.: Decidenet: Counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5197–5206.IEEE, Piscataway, NJ (2018)
26. Kang, D., Chan, A.: Crowd counting by adaptively fusing predictions from an image pyramid. arXiv:180506115 (2018)
27. Lin, G., et al.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925–1934.IEEE, Piscataway, NJ (2017)
28. Chen, L.C., et al.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. 40(4), 834–848 (2018)
29. Deb, D., Ventura, J.: An aggregated multicolor dilated convolution network for perspective-free counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 195–204.IEEE, Piscataway, NJ (2018)
30. Sam, D.B., et al.: Almost unsupervised learning for dense crowd counting. Proceedings of the AAAI Conference on Artificial Intelligence 33(1), 8868–8875 (2019)
31. Liu, X., van de Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7661–7669.IEEE, Piscataway, NJ (2018)
32. Walach, E., Wolf, L.: Learning to count with cnn boosting. In: European conference on computer vision, pp. 660–676.Springer, Berlin (2016)
33. Ranjan, V., Le, H., Haoi, M.: Iterative crowd counting. In: Proceedings of the European Conference on Computer Vision, pp. 270–285.Springer, Berlin (2018)
34. Liu, J., et al.: Crowd counting using deep recurrent spatial-aware network. arXiv:180700601 (2018)
35. Jaderberg, M., et al.: Spatial transformer networks. In: Advances in neural information processing systems, pp. 2017–2025.MIT Press, Cambridge, MA (2015)
36. Sindagi, V.A., Patel, V.M.: Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: 2017 14th IEEE
Ji Q, Chen H, Bao D. Improving crowd counting with scale-aware convolutional neural network. IET Image Process. 2021;15:2192–2201.

https://doi.org/10.1049/ipr2.12187