2019 IEEE Winter Conference on Applications of Computer Vision

Scale Pyramid Network for Crowd Counting

Xinya Chen Yanrui Bin Nong Sang* Changxin Gao
Key Laboratory of Ministry of Education for Image Processing and Intelligent Control,
School of Automation,Huazhong University of Science and Technology
{hust_cxy,yrbin,nsang,cgao}@hust.edu.cn

Abstract

Crowd counting is a concerned yet challenging task in computer vision. The difficulty is particularly pronounced by scale variations in crowd images. Most state-of-art approaches tackle the multi-scale problem by adopting multi-column CNN architectures where different columns are designed with different filter sizes to adapt to variable pedestrian/object sizes. However, the structure is bloated and inefficient, and it is infeasible to adopt multiple deep columns due to the huge resource cost. We instead propose a Scale Pyramid Network (SPN) which adopts a shared single deep column structure and extracts multi-scale information in high layers by Scale Pyramid Module. In Scale Pyramid Module, we specifically employ different rates of dilated convolutions in parallel instead of traditional convolutions with different sizes. Compared to other methods of coping with scale issues, our single column structure with Scale Pyramid Module can get more accurate estimation with simpler structure and less complexity of training. And our Scale Pyramid Module can be easily applied to a deep network. Experimental results on four datasets show that our method achieves ***state-of-the-art*** *performance. On ShanghaiTech Part_A dataset which is challenging for its highly congested scenes and scale variation, we achieve 9.5% lower MAE and 13.5% lower MSE than the previous state-of-the-art method. We also extend our model on TRANCOS vehicle counting dataset and significantly achieve 5.9% lower GAME(0), 10% lower GAME(1), 24.5% lower GAME(2), 38.7% lower GAME(3) than the previous state-of-the-art method. The experimental results prove the robustness of our model for crowd counting, especially with scale variations.*

1. Introduction

Crowd counting has gained an increasing interest for its applications in video surveillance [3, 8], public safety, flow monitoring, traffic monitoring, and scene understanding. It is a challenging task due to various cases such as intra-scene and inter-scene variations in scale, occlusions, non-uniform distribution, illumination variation. In this paper, we propose a novel architecture mainly coping with the scale variations, as shown in Fig. 1, which significantly influence the performance.

Figure 1. Samples in the ShanghaiTech Part_A dataset [40].The scale varies significantly within the scene and between scenes.

Previous approaches basically adopt three kinds of methods to address this issue, *i.e*. multi-column-based methods, image pyramid-based methods, and multi-level feature-based methods, as shown in Fig. 2(a)(b)(c). Recently most state-of-art work adopt the first kind of methods which employ multi-column CNN architectures [7, 20, 23, 31, 40] where different columns are designed with different filter sizes to adapt to variable scales. They extract multi-scale features from the original images in early layers and then process the features of particular scale respectively. Although this kind of methods has shown robust performance, they still have two significant disadvantages mainly caused by the multi-column structure and the large-sized filters. First, the multi-column structure is bloated and increases the number of parameters, which are also increased exponentially when using large-sized filters in the column. The

*Corresponding author.

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00211

Method	Parameters	MAE	MSE
MCNN	127.68k	110.2	185.9
Col.3 of MCNN-SPM-C	55.04k	95.0	139.9
Col.3 of MCNN-SPM-A	44.68k	94.2	150.2

Table 1. Comparative experiments on ShanghaiTech Part_A [40]. The architecture of Col.3 of MCNN-SPM-C/A is: CR(24,5)-M-CR(48,3)-M-CR(24,3)-(SPM-C/A)-CR(12,3)-CR(1,1). CR(m,n) means the convolution layer with m filters whose size is $n \times n$ followed by the ReLu layer. M is the max pooling layer.

increased parameters lead to the increase of training time and computation amount. Secondly, the structure needs the pre-training of every single column according to the training strategy described in [40], which is complex and also leads to the increase of training time. Moreover, the large-sized filters make the network hard to train. The above disadvantages make it infeasible to apply the multi-column structure to a deep network, while a deeper network has been proved to have better performance in crowd counting [18].

By observing the feature maps generated by Multi-Column Convolutional Neural Network (MCNN) [40], we found the low-level features from the same depth of different columns are similar. Hence, we adopt only one column of MCNN and extract multi-scale features from the high-level feature (as shown in Fig. 2 (d)). Moreover, we discard the large-sized filters. When extracting multi-scale features, we design a Scale Pyramid Module (SPM) to deliver multiple receptive fields with fewer parameters. The module employs multiple parallel dilated convolutions with different rates instead of different sizes of traditional convolutions.

Specifically, we select the third column which adopts small kernels as our backbone and embed the SPM, which employs four parallel dilated convolutions with rates as 2, 4, 8, 12, between the third convolution and the forth convolution. Each dilated convolution in SPM has the same number of channels as the input features. Then the features generated by SPM merged together by concatenation or addition. We denoted the model with concatenation as Col.3 of MCNN-SPM-C, and the model with addition as Col.3 of MCNN-SPM-A. We conduct experiments on ShanghaiTech Part_A dataset [40]. The results are shown in Table 1. Compared to MCNN, both the two models perform significantly better with significantly fewer amounts of parameters.

We further apply the method to a deeper network to obtain a better performance. We choose VGG-based structure as our backbone and construct the Scale Pyramid Network (SPN). We conduct experiments on four public datasets. Our SPN outperforms the previous state-of-art approach called CSRNet [18] with 9.5%, 11.3%, 2.6% lower mean absolute error (MAE), 13.5%, 10.0%, 15.5% lower mean square error (MSE) on ShanghaiTech [40]Part_A, Part_B, UCF_CC_50 datasets [10] respectively. Moreover, we extend our method to vehicle counting on the TRANCOS dataset [24] and achieve 5.9% lower GAME(0), 10% lower GAME(1), 24.5% lower GAME(2), 38.7% lower GAME(3) than the state-of-art result generated by CSRNet.

Compared to other methods of coping with scale issues, our single column structure with Scale Pyramid Module is more efficient, more effective and easier to train. The introduced Scale Pyramid Module enhances the robustness against scale variation with only a small increase in complexity and can be applied to the deep structure.

2. Related Work

Various methods have been proposed for crowd counting and density estimation [2, 13, 21]. Most of the early researches adopted detection-based methods using a moving-window-like detector to detect people and count the number [25]. These methods were easily affected by occlusions and high clutter in the background. Recent methods can be classified into three categories: Regression-based methods, density estimation-based methods, and CNN-based methods. Here we mainly review the three categories of methods.

2.1. Regression-Based Methods

To address the issues of occlusion and clutter, researchers try to deploy regression-based methods to learn a mapping from various features extracted from local image patches to object counts [4, 12]. They fist extract low-level feature and then perform regression modeling. Handcrafted features, like edge features and texture features [1, 12], are used to generate low-level information. For example, Idrees *et al*. [9] proposed a model to extract features by employing Fourier analysis and SIFT interest-point.

2.2. Density Estimation-Based Methods

Regression-based methods performed well in tackling the occlusion and clutter problems. However, they ignored the spatial information due to the regression to one count. Hence, Lempitsky *et al*. [17] introduced a new method which learns a linear mapping between local region features and corresponding object density maps by regression. Since it is difficult to learn an ideal linear mapping, Pham *et al*. [33] proposed a method which uses random forest regression to learn a non-linear mapping. After that, many approaches adopt density map regression for crowd counting [26, 32, 36, 38].

2.3. CNN-Based Methods

Recently, the success of convolutional neural network(CNN) in computer vision has inspired researchers to apply them to density estimation. According to the way tackling the multi-scale problems, we divide the existing

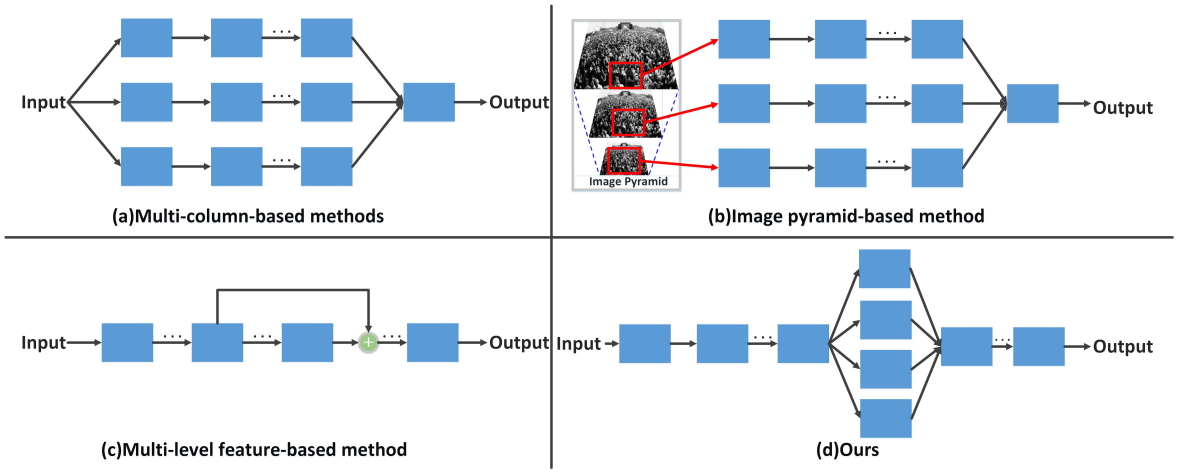

Figure 2. Overview of existing methods coping with scale changes. (a) Multi-column-based methods. (b) Image pyramid-based methods. (c) Multi-level feature-based methods. (d) Our method.

methods into three categories as multi-column-based methods, image pyramid-based methods, multi-level feature-based methods, as shown in Fig. 2 (a)(b)(c).

(a)Multi-column-based methods: This method adopts multiple columns which have different receptive fields by adopting different sizes of filters to adapt to variable target sizes. Typical examples include Zhang *et al*. [40] who adopt three columns with small, medium, large kernels respectively and merge them in the end to generate the density map. Based on Zhang's architecture, Sam *et al*. [7]proposed the switch-CNN which uses a density level classifier to choose the suitable column for particular input patches. And Sindagi *et al*. [31] proposed the CP-CNN which adds an image-wise density level classifier to get global context information for achieving lower count error, and a patch-wise density level classifier to get local context information for achieving better quality density map. Most recently, Deb *et al*. [5] incorporate dilated filters into the multi-column structure, where different columns adopt different rates of dilation convolutions.

(b)Image pyramid-based methods: This method crops patches from the multi-scale pyramidal representation of each image as inputs to provide the multi-scale information to the learnable network. The patches from small scaled images contain more global information, while patches from large scaled images preserve more details. Typical examples include $\tilde{O}noro - Rubio$ *et al*. [23] who proposed the HydraCNN which present differently scaled patches of one image to the corresponding column and then merge the features to incorporate multi-scale, global information. Also Boominathan *et al*. [20] adopt the image pyramid based method for data argumentation. The main drawback of these methods comes at the cost of computing features responses at all layers for multiple scaled versions of the input images.

(c)Multi-level feature-based methods: This method extract features of multiple levels to utilize information of different characteristics or different scales. For example, Boominathan *et al*. [20] utilizes a shallow network to recognize the low-level head blob patterns which are arisen from people away from the camera and a deep network to capture the details of people near the camera. The two columns utilize the different characteristics of different scales respectively to cope with the intra-scene scale changes. In the end, the multi-level features are merged to map the density map. Zhang *et al*. [16] combine the features from multiple layers of a single column structure, which based on the method that different layers have different receptive fields corresponding different scales. However, the improvement is not obvious in congested scenes, where the scale changes exist significantly. This kind of method is widely used in semantic segmentation and target detection task. However, it may not suitable for crowd counting for the reason that low-level feature contains more edge information, which may disturb the estimation due to the non-uniform distribution of edge caused by scale changes.

3. Proposed Method

3.1. SPN Architecture

Most state-of-art approaches address multi-scale issues by adopting multi-column architectures with different filter sizes. They extract multi-scale features from original images in early layers and then process particular scale of features respectively. Finally, the multi-scale features are merged together to map to the density map. However, the structure is bloated and inefficient. It is infeasible to deploy multiple deep columns due to the huge resource

cost including training time, parameter numbers, memory consumption and computation amount. While a deep network has been proved to have a good performance [18] in crowd counting. By observing the feature maps generated by MCNN [40], we found the low-level features from the same depth of different columns are similar, which encode low-level spatial visual information like edges, circles, etc. Therefore, we employ a single column structure as shared backbone and extract multi-scale features from high-level features in high layers, the experiment described in **Introduction** has shown that the method is more effective and efficient with fewer parameters than multi-column-based method.

We adopt VGG-16 [29]-based network as our backbone for its strong transfer learning ability. The original VGG-16 network have five pooling layers which can enlarge the receptive field and reduce the amount of computation, but leads to the loss of spatial information. Considering the tradeoff between accuracy and resource cost, we remove the last two pooling layers. To extract high-level features of multiple scales, we design a Scale Pyramid Module and embed the module between conv4_3 and conv5_1. Experiments show that deployment with the location between conv4_3 and conv5_1 performs better than it between conv3_3 and conv4_1. Furthermore, We remove the three fully-connected layer to make the structure adapt to the inputs of arbitrary resolution and adopt a 1×1 convolution to map to the density map. Note that, we adopt the Rectified linear unit (ReLU) activation function after the last convolution to make sure that the estimated value is not less than zero.The architecture of SPN is: $2 \times CR(64, 3)$-M-$2 \times CR(128, 3)$-M-$3 \times CR(256, 3)$-M-$3 \times CR(512, 3)$-SPM-$3 \times CR(512, 3)$-CR(256,3)-CR(1,1). $N \times CR(m, n)$ means N convolution layers with m filters whose size is $n \times n$ followed by the ReLu layer. If N is 1, $1\times$ will be omitted. M is the max pooling layer.

3.2. Scale Pyramid Module

We design the Scale Pyramid Module to extract high-level features of multiple scales. Adopting convolutions of multiple sizes in parallel is a feasible method. However, the number of parameters increases as larger kernels are used to extract features at larger scales. Inspired by [14, 19], we replace the traditional convolutions of multiple sizes with the dilated convolutions which can obtain different receptive fields when adopting different rates (as shown in Fig. 3(b)). Unlike the traditional convolution, the dilated convolution doesn't increase the number of parameters and the amount of computation while expanding the receptive field.

The dilated convolution is first proposed by Yu *et al*. [41] in segmentation task. It is a generalization of the traditional convolution which introduces some 'holes' to skip part of the input. Let $F : \mathbb{Z}^2 \to \mathbb{R}$ be a discrete function. Let $\Omega_m = [-m, m]^2 \cap \mathbb{Z}^2$ and let $k : \Omega_m \to \mathbb{R}$ be a discrete filter of size $(2m + 1)^2$. The traditional convolution operation can be defined as:

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t) \, . \tag{1}$$

Let r be the dilation rate,the dilated convolution operation can be defined as:

$$(F *_r k)(p) = \sum_{s+rt=p} F(s)k(t) \, . \tag{2}$$

It can be seen that a $k \times k$ size of traditional convolution can be enlarged to $k + (k - 1)(r - 1)$ without increasing the number of parameters by introducing r-1 zeros between consecutive filter values.

In our Scale Pyramid Module, we employ four parallel dilated convolutions with dilated rate as 2, 4, 8, 12 respectively (as shown in Fig. 3). Each dilated convolution in SPM has the same number of channels as the input features. Given input features maps, the four dilated convolutions extract multi-scale features with the receptive field of $5 \times 5, 9 \times 9, 17 \times 17, 25 \times 25$. Then the multi-scale features are merged together and fed to the following convolution layer for further processing. Moreover, we also merge the input features in order to adapt to more scales. The features can be merged by concatenation or addition. Experiments in **Introduction** show that concatenation can get similar MAE but better MSE than addition. In order to achieve good performance both in MAE and MSE, we merge the features by concatenation in the following experiments. The merged features with different receptive fields capture the targets at different scales, thus enhance the robustness of the model against scale variations.

3.3. Ground Truth Density Maps

We generate the density maps following the previous work [40, 18]. For each pedestrian head at pixel x_i, we represent it by a delta function $\delta(x - x_i)$. To obtain a continuous density function, the ground truth density map $F(x)$ is computed by convolving the delta function with a Gaussian Kernel G_{σ_i} normalized to 1, which is defined as:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_{\sigma_i}(x) \, . \tag{3}$$

The value of σ_i is set by considering the crowd distribution of all the images in the dataset. For the UCF_CC_50 dataset which suffered significant scale changes, we use the geometry-adaptive kernels following the method of generating density maps in [40], the σ_i of head x_i is determined by the average distance $\overline{d}_i$ of k nearest neighbors, which is defined as follow:

$$\sigma_i = \beta \overline{d}_i \, . \tag{4}$$

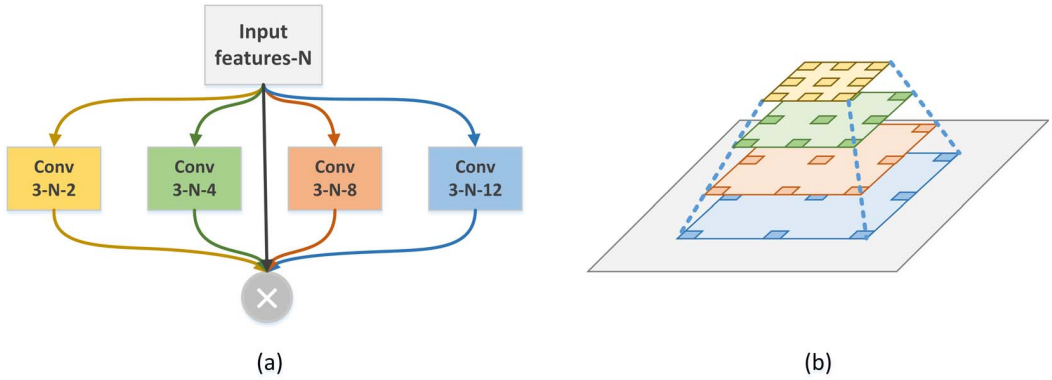

Figure 3. (a) The structure of our Scale Pyramid Module.The convolution layer's parameters are denoted as "conv-(kernel size)-(number of filters)-(dilation rate)". All convolution layers use padding to maintain the previous size and followed by the ReLu layer. $\bigotimes$ denotes the fusion operation which can be achieved by concatenation or addition. (b) The dilated convolutions capturing targets at different scales concatenate like a pyramid, thus we call the module as Scale Pyramid Module.

Dataset	Generating method
ShanghaiTech Part_A	$\sigma_i = 4$
ShanghaiTech Part_B	$\sigma_i = 15$
UCF_CC_50	Geometry-adaptive kernels
the UCSD	$\sigma_i = 3$
TRANCSO	$\sigma_i = 10$

Table 2. The ground truth generating methods for different datasets.

Here we follow the configuration in [40] with $\beta = 0.3$ and $k = 2$. For the other datasets, we simply adopt σ_i with the fixed value following [18]. Specific settings are shown in the Table 2. Since we use three max pooling layers, we generate the ground truth density map with the size 1/64 of input size.

4. Experiments

4.1. Training Details

We conduct the experiments on three crowd counting datasets [1, 10, 40] and a vehicle counting dataset [24]. For each image in the training set, we augment it by randomly cropping 9 patches with 1/4 size of the original image, and then flipping each patch in the horizontal direction. We implement our model based on the Caffe framework designed by Jia *et al*. [39]. In all experiments, we set the batch size as 1 and employ stochastic gradient descent(SGD) as the optimization with the momentum at 0.9. And we adopt a fixed learning rate with different values on different datasets. To cope with the overfitting, we employ L2 regularization with the weight decay at 0.0005. The layers introduced from the VGG-16 structure are fine-tuned from a pre-trained model. The other layers adopt Gaussian initialization with 0.01 standard deviation. Following the work [40, 20, 23], we also adopt the Euclidean loss to measure the distance between the ground truth and the estimated density map. The loss function is given as follow:

$$L(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|F(X_i;\Theta) - F_i\|_2^2 \,. \tag{5}$$

Where Θ refers to the set of learnable parameters in the SPN. X_i is the input image. $F(X_i;\Theta)$ denotes the estimated density map generated by SPN for image X_i. F_i is the corresponding ground truth density map of image X_i. N is the number of training images. L is the loss between the ground truth density map and the estimated density map.

4.2. Evaluation Metrics

Following the previous works [2, 7, 20, 40], we evaluate the performance via the mean absolute error (MAE) and mean square error (MSE) which are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |z_i - \hat{z}_i| \,. \tag{6}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |z_i - \hat{z}_i|^2} \,. \tag{7}$$

where N is the number of test images, z_i represents the actual number of people in the ith image, and $\hat{z}_i$ represents the estimated count in the ith image. The estimated count is calculated by integrating the estimated density map. Roughly speaking, MAE indicates the accuracy of the estimation, and MSE indicates the robustness of the estimation [40].

	Part_A		Part_B	
Method	MAE	MSE	MAE	MSE
Zhang *et al*. [2]	181.8	277.7	32.0	49.8
Marsden *et al*. [22]	126.5	173.5	23.8	33.1
MCNN[40]	110.2	173.2	26.4	41.3
Cascaded-MTL [30]	101.3	152.4	20.0	31.1
Switching-CNN [31]	90.4	135.0	21.6	33.4
DecideNet [11]	-	-	20.75	29.42
SaCNN [16]	86.8	139.2	16.2	25.8
ACSCP [28]	75.7	102.7	17.2	27.4
CP-CNN [31]	73.6	106.4	20.1	30.1
IG-CNN [6]	72.5	118.2	13.6	21.1
Liu *et al*. [37]	72.0	106.6	14.4	23.8
ic-CNN [34]	68.5	116.2	10.7	16.0
CSRNet [18]	68.2	115.0	10.6	16.0
Ours(SPN)	**61.7**	**99.5**	**9.4**	**14.4**

Table 3. Estimation errors on ShanghaiTech dataset.

4.3. ShanghaiTech Dataset

ShanghaiTech dataset [40] is a large-scale crowd counting dataset which consists of 1198 annotated images with a total of 330,165 people. This dataset consists of two Parts: Part_A includes 482 images in highly congested scenes with counts ranging from 33 to 3139, while Part_B includes 716 images in relatively sparse scenes with counts ranging from 9 to 578. Following [40], we use 300 images for training and 182 images for testing in Part_A, 400 images for training and 316 images for testing in Part_B. We train the model following the methods given in section 4.1 with the learning rate at 10^{-7} on Part_A and 10^{-6} on Part_B.

The results of our method and previous state-of-art methods are shown in the Table 3. Results show that our method achieves the lowest MAE and MSE both on Part_A and Part_B. The previous state-of-art method called CSRNet [18] is a single column structure which deploys a convolutional neural network as the front-end for feature extraction and a dilated CNN for the back-end to deliver larger reception fields. Compared to CSRNet, our SPN model can extract more multi-scale features, and we achieve 9.5% lower MAE and 13.5% MSE on Part_A, which demonstrates the effectiveness of our SPN model and its robustness in multi-scale scenes. On Part_B, our model also achieves 11.3% lower MAE and 10.0% lower MSE than CSRNet, which indicates that our model can perform well not only in extremely dense scenes but also in relative sparse scenes.

4.4. Ablation Study on ShanghaiTech Part_A

In order to analyze the effectiveness of our multi-scale features extraction module, we conduct an ablation study on the ShanghaiTech Part_A [40] dataset.

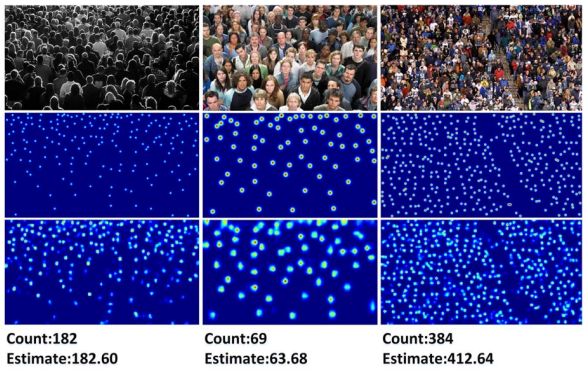

Figure 4. Examples on ShanghaiTech Part_A [40] dataset. The first row shows images in testing set. The second row shows the corresponding ground truth. The third row shows the generated density map.

Method	MAE	MSE
SRN	68.8	114.7
VGG-16 Backbone	66.7	110.8
Ours(SPN)	61.7	99.5

Table 4. Ablation study on ShanghaiTech Part_A.

Scale ratio		0.8	0.9	1.1	1.2	Mean	Std
VGG-16	MAE	73.7	69.8	72.6	76.4	73.1	2.37
VGG-16	MSE	139.6	128.2	124.8	128.7	130.3	5.56
SPN	MAE	70.1	66.0	66.5	68.8	67.8	1.67
SPN	MSE	122.1	110.6	111.0	114.7	114.6	4.62

Table 5. Comparative experiments about sensitivity to scale change. VGG-16 Backbone is omitted as VGG-16.

We remove our Scale Pyramid Module, that is VGG-16 [29] model except for the fully connected layers and two max pooling layers. Then we train it using the same training method with the fixed learning rate at 10-7. We denote this model as VGG-16 Backbone. As shown in Table 4, VGG-16 Backbone gets the MAE of 66.7 and the MSE of 110.8, while our SPN model obtains the MAE of 61.7 and the MSE of 99.5. Our model achieves 7.5% lower MAE and 10.2% lower MSE, which demonstrates that the Scale Pyramid Module can significantly decrease the error of estimated crowd count in congested scenes with varied scales. To further compare the two models' sensitive to scale changes, we test the VGG-16 Backbone and SPN using different scale ratio on the test set. Specifically, we resize the original image during test with a fixed ratio as 0.8, 0.9, 1.1, 1.2 and utilize standard deviation of MAE and MSE to measure the sensitivity to scale change. Results are shown as Table 5. Our SPN model gets the smaller standard deviation of MAE and MSE, which demonstrates that the Scale Pyramid Module can decrease the sensitivity to scale change.

Configuration	2	2-4	2-4-8
MAE	66.9	64.6	62.3
MSE	107.6	108.6	99.5

Table 6. Experiments about different number of branches in Scale Pyramid Module.

Method	Parameters	MAE	MSE
MCNN	127.68k	110.2	185.9
MCNN-SPM	164.58k	102.6	147.6
MCNN-MSPM	227.49k	98.1	143.3

Table 7. Comparative experiments on ShanghaiTech Part_A [40].

However, the increased number of parameters introduced by our Scale Pyramid Module may also improve the performance by increasing the capacity of the model. For the purpose of studying the impact of increased parameters on regression results, we replace all the dilated convolutions with 3×3 convolutions which have the same number of parameters as the dilated convolutions. Then we train it with the fixed learning rate at 10^{-6}. We denote this model as Same Rate Network(SRN). To our surprise, the performance of SRN is worse than the VGG-16 Backbone, with the MAE of 68.8 and the MSE of 114.7. The results demonstrate the effectiveness of the structure with parallel dilated convolutions at different rates.

In order to analyze the impact of the number of branches in Scale Pyramid Module. We design three kind of Scale Pyramid Modules which respectively consists of one branch with dilation rate at 2, two branches with dilation rate at 2 and 4, three branches with dilation rate at 2,4,8. We train them with the fixed learning rate at 10-7. The detailed evaluation results are shown in Table 6. The increase of branch can improve the performance but will tend to saturate later.

We also apply the SPM module to multi-column structure and experiment on ShanghaiTech Part_A dataset.We apply the module to MCNN by two ways. For the first way, we embed SPM to each column of MCNN, each SPM is between the third convolution and the forth convolution of the column. We denote this model as MCNN-MSPM. For the second way, we embed SPM after the fusion operation and add a 1×1 convolution with 30 channels before the last convolution to decrease the channels smoothly. We denote this model as MCNN-SPM. Each dilated convolution in SPM has the same number of channels as the input features. The results in Table 7 shows that the two model both perform better than MCNN, which demonstrates the effectiveness of SPM.

4.5. The UCF_CC_50 Dataset

The UCF_CC_50 dataset [10] contains 50 images in extremely congested scenes. The counts range from 94 to 4543 with an average of 1280 individuals per image. It is an extremely challenging dataset due to the small dataset size, large variance in crowd count, congested scenes and large-scale change. Following the work of [10], we perform 5-fold cross validation on this dataset. Due to the small dataset size, we fixed the first ten layers and only train the other layers. In different training sets, we adopt different learning rates.

Method	MAE	MSE
Idrees *et al*. [9]	419.5	541.6
Zhang *et al*. [2]	467.0	498.5
MCNN [40]	377.6	509.1
Onoro *et al*. [23] Hydra-2s	333.7	425.2
Onoro *et al*. [23] Hydra-3s	465.7	371.8
Walach *et al*. [35]	364.4	341.4
Marsden *et al*. [22]	338.6	424.5
Cascaded-MTL [30]	322.8	397.9
Switching-CNN [7]	318.1	439.2
SaCNN [16]	314.9	424.8
CP-CNN [31]	295.8	**320.9**
ACSCP [28]	291.0	404.6
IG-CNN [6]	291.4	349.4
AMDCN [5]	290.82	-
Liu *et al*. [37](Keyword)	279.6	388.9
CSRNet [18]	266.1	397.5
ic-CNN [34]	260.9	365.5
Ours(SPN)	**259.2**	335.9

Table 8. Estimation errors on UCF_CC_50 dataset.

Our method is evaluated and compared with previous state-of-art methods. The results are summarized in Table 8, it can be seen that our model achieves the lowest MAE and MSE. Compared to CSRNet [18], we achieve 2.6% lower MAE and 15.5% MSE, which indicates the robustness of our method against the scale changes and the effectiveness of our simple model.

4.6. The UCSD Dataset

The UCSD dataset [1] contains 2000 frames captured from a stationary digital camcorder overlooking a pedestrian walkway at UCSD. They are in sparse scenes with counts ranging from 11 to 46 per image, and totally contains 49,885 pedestrians. The region of interest(ROI) is provided for the whole dataset so that the crowd counting is only conducted in the ROI. Following the same setting with [1], we use frames from 601 to 1400 as training data and the remaining 1200 frames as test data. This split tests the generalization ability and robustness of the crowd-counting system. Before training, we first resize each frame from size 158×238 to 952×632 to tackle the reduction of resolution due to the pooling operations. Then we set the intensities of pixels out of ROI in the frame and the corresponding area

Method	MAE	MSE
Zhang *et al*. [2]	1.60	3.31
CCNN [23]	1.51	-
Switching-CNN [7]	1.62	2.10
FCN-rLSTM [27]	1.54	3.02
CSRNet [18]	1.16	1.47
MCNN [40]	1.07	1.35
ACSCP [28]	1.04	1.35
Ours(SPN)	**1.03**	**1.32**

Table 9. Estimation errors on UCSD dataset.

Method	GAME 0	GAME 1	GAME 2	GAME 3
Fiaschi *et al*. [15]	17.77	20.14	23.65	25.99
Lempitsky *et al*. [17]	13.76	16.72	20.72	24.36
Onoro *et al*. [23] Hydra-3s	10.99	13.75	16.69	19.32
AMDCN [5]	9.77	13.16	15.00	15.87
FCN-HA [27]	4.21	-	-	-
CSRNet [18]	3.56	5.49	8.57	15.04
Ours(SPN)	**3.35**	**4.94**	**6.47**	**9.22**

Table 10. GAME on TRANCOS dataset.

in the density map to zero. The training method is given in section 4.1 and we set the learning rate at 10^{-6}.

Table 9 shows the results of our model and previous models. The results show that our model achieves state-of-the-art performance, which indicates that our model can estimate images not only with extremely dense crowds but also with relative sparse people.

4.7. The TRANCOS Vehicle Counting Dataset

Besides the crowd counting datasets, we further conduct an experiment on TRANCOS vehicle counting dataset which consists of 1244 traffic jam images with totally 46796 vehicles captured by real traffic surveillance cameras. This dataset covers a variety of different scenes and viewpoints. All the collected images contain traffic congestion with changes in lighting conditions, different levels of overlap and crowdedness, even within the same image. The region of interest (ROI) to identify the road region is provided for each image.

This dataset introduces a new metric called Grid Average Mean Absolute Error (GAME) to provide a more accurate evaluation. This metric simultaneously considers the object count and the location estimated for the objects. The GAME(L) splits a given density map into 4^L non-overlapping regions and compute the MAE in each of these sub-regions. Then these individual errors are summed to obtain the final GAME for a particular image. The GAME is formulated as follows:

$$GAME(L) = \frac{1}{N}\sum_{n=1}^{N}(\sum_{l=1}^{4^L}|e_n^l - gt_n^l|)\ . \qquad (8)$$

We train the model with learning rate at 10^{-5}. Table 10 shows the results of our model and previous models.

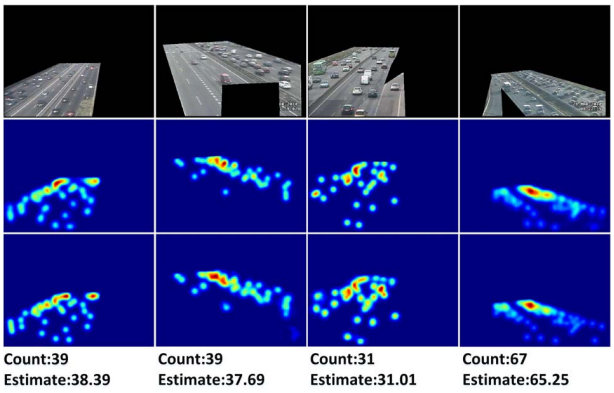

Figure 5. Examples in TRANCOS [24] dataset. The first row shows images in testing set. The second row shows the corresponding ground truth. The third row shows the generated density map.

Our model achieves significant improvement, especially on higher L of GAME metrics. Compared to the result of Onoro *et al*. [23] method which employs an MCNN-like architecture, our method achieves 69.5% lower GAME(0), 64.1% lower GAME(1), 61.2% lower GAME(2), 52.3% lower GAME(3). Compared to CSRNet, our method achieve 5.9% lower GAME(0), 10% lower GAME(1), 24.5% lower GAME(2), 38.7% lower GAME(3). It can be seen that our method can be extended to other counting tasks and also performs well, which indicates the great robustness and generalization of our model. And the lower GAME indicates that our model not only achieves a lower error in object count but also has a more accurate distribution.

5. Conclusions

In this paper, we propose a novel architecture called Scale Pyramid Network (SPN) for crowd counting. We use a single column structure as the backbone and extract high-level features of multiple scales by dilated convolutions with different rates in parallel. Our method can easily adapt to multi-scale scenes with simpler structure and less complexity of training. Extensive experiments are conducted on the challenging crowd counting datasets and our model gets the significantly better performance against the state-of-art methods, which demonstrates the efficiency of our method. What's more, we extend our model to vehicle counting task and also achieve the best performance.

Acknowledgement

This work was supported by National Key R&D Program of China (No.2018YFB1004600), and the Fundamental Research Funds for the Central Universities No.2017KFYXJJ179.

References

[1] Z.-S. J. L. A. B. Chan and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[2] X. W. C. Zhang, H. Li and X. Yang. Cross-scene crowd counting via deep convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 83–841, 2015.

[3] Z. M. D. Kang and A. B. Chan. Beyond counting: Comparisons of density maps for crowd analysis tasks-counting,detection, and tracking. *IEEE Transactions on Circuits Systems for Video Technology*, 2017.

[4] C. F. D. Ryan, S. Denman and S. Sridharan. Crowd counting using multiple local features. *Digital Image Computing: Techniques and Applications*, pages 81–88, 2009.

[5] D. Deb and J. Ventura. An aggregated multicolumn dilated convolution network for perspective-free counting. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[6] R. V. B. Deepak Babu Sam, Neeraj N Sajjan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] S. S. Deepak Babu Sam and R. V. Babu. Switching convolutional neural network for crowd counting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1:6, 2017.

[8] X. S. F. Xiong and D.-Y. Yeung. Spatiotemporal modeling for crowd counting in videos. *IEEE International Conference on Computer Vision*, pages 5161–5169.

[9] C. S. H. Idrees, I. Saleemi and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013.

[10] C. S. Haroon Idrees, Imran Saleemi and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013.

[11] C. G. Jiang Liu and D. Meng. Decidenet: Counting varying density crowds through attention guided detection and density estimation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[12] S. G. K. Chen, C. C. Loy and T. Xiang. Feature mining for localised crowd counting. *European Conference on Computer Vision*, 2012.

[13] T. X. K. Chen, S. Gong and C. C. Loy. Cumulative attribute space for age and crowd density estimation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2467–2474, 2013.

[14] I. K. K. M. L. C. Chen, G. Papandreou and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[15] R. N. L. Fiaschi, U. Koethe and F. A. Hamprecht. Learning to count with regression forest and structured labels. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 2685–2688, 2012.

[16] Q. C. L Zhang, M Shi. Crowd counting via scale-adaptive convolutional neural network. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[17] V. Lempitsky and A. Zisserman. Learning to count objects in images. *Advances in Neural Information Processing Systems*, pages 1324–1332, 2010.

[18] Y. Li, X. Zhang, and D. Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] F. S. Liang-Chieh Chen, George Papandreou and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, 2017.

[20] S. S. K. Lokesh Boominathan and R. V. Babu. Crowdnet: a deep convolutional network for dense crowd counting. *Proceedings of the 2016 ACM on Multimedia Conference*, pages 640–644, 2016.

[21] J. S. M. Rodriguez, I. Laptev and J.-Y. Audibert. Density-aware person detection and tracking in crowds. *2011 International Conference on Computer Vision*, pages 2423–2430, 2011.

[22] S. L. Mark Marsden, Kevin McGuiness and N. E. OConnor. Fully convolutional crowd counting on highly congested scenes. *arXiv preprint arXiv:1612.00220*, 2016.

[23] D. Onoro-Rubio and R. J. Lpez-Sastre. Towards perspective-free object counting with deep learning. *European Conference on Computer Vision*, pages 615–629, 2016.

[24] D. Onoro-Rubio and R. J. Lopez-Sastre. Towards perspective-free object counting with deep learning. *European Conference on Computer Vision*, pages 615–629, 2016.

[25] B. S. Piotr Dollar, Christian Wojek and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.

[26] S. A. S. S. A. M. Saleh and H. Ibrahim. Recent survey on crowd density estimation and counting for visual surveillance. *Engineering Applications of Artificial Intelligence*, 41:103–114, 2015.

[27] J. P. C. Shanghang Zhang, Guanhang Wu and J. M. Moura. Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3667–3676, 2017.

[28] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang. Crowd counting via adversarial cross-scale consistency pursuit. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[30] V. A. Sindagi and V. M. Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. *Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017.

[31] V. A. Sindagi and V. M. Patel. Generating highquality crowd density maps using contextual pyramid cnns. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2017.

[32] V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 2017.

[33] O. Y. V.-Q. Pham, T. Kozakaya and R. Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. *Proceedings of the IEEE International Conference on Computer Vision*, pages 3253–3261, 2015.

[34] M. H. Viresh Ranjan, Hieu Le. Iterative crowd counting. *arXiv preprint arXiv:1807.09959*, 2018.

[35] E. Walach and L. Wolf. Learning to count with cnn boosting. *European Conference on Computer Vision*, pages 660–676, 2016.

[36] Y. Wang and Y. Zou. Fast visual object counting via example-based density estimation. *2016 IEEE International Conference on Image Processing(ICIP)*, pages 3653–3657, 2016.

[37] A. D. B. Xialei Liu, Joost van de Weijer. Leveraging unlabeled data for crowd counting by learning to rank. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[38] B. Xu and G. Qiu. Crowd density estimation based on rich features and random projection forest. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8, 2016.

[39] J. D. S. K. J. L. R. G. S. G. Yangqing Jia, Evan Shelhamer and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.

[40] S. C. S. G. a. Y. M. Yingying Zhang, Desen Zhou. Single-image crowd counting via multi-column convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016.

[41] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.