CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes

Yuhong Li[1,2], Xiaofan Zhang[1], Deming Chen[1]
[1]University of Illinois at Urbana-Champaign
[2]Beijing University of Posts and Telecommunications
{*leeyh,xiaofan3,dchen*}*@illinois.edu*

Abstract

We propose a network for Congested Scene Recognition called CSRNet to provide a data-driven and deep learning method that can understand highly congested scenes and perform accurate count estimation as well as present high-quality density maps. The proposed CSRNet is composed of two major components: a convolutional neural network (CNN) as the front-end for 2D feature extraction and a dilated CNN for the back-end, which uses dilated kernels to deliver larger reception fields and to replace pooling operations. CSRNet is an easy-trained model because of its pure convolutional structure. We demonstrate CSRNet on four datasets (ShanghaiTech dataset, the UCF_CC_50 dataset, the WorldEXPO'10 dataset, and the UCSD dataset) and we deliver the state-of-the-art performance. In the ShanghaiTech Part_B dataset, CSRNet achieves 47.3% lower Mean Absolute Error (MAE) than the previous state-of-the-art method. We extend the targeted applications for counting other objects, such as the vehicle in TRANCOS dataset. Results show that CSRNet significantly improves the output quality with 15.4% lower MAE than the previous state-of-the-art approach.

1. Introduction

Growing number of network models have been developed [1, 2, 3, 4, 5] to deliver promising solutions for crowd flows monitoring, assembly controlling, and other security services. Current methods for congested scenes analysis are developed from simple crowd counting (which outputs the number of people in the targeted image) to density map presenting (which displays characteristics of crowd distribution) [6]. This development follows the demand of real-life applications since the same number of people could have completely different crowd distributions (as shown in Fig. 1), so that just counting the number of crowds is not enough. The distribution map helps us for getting more accurate and comprehensive information, which could be critical for making correct decisions in high-risk environments, such as stampede and riot. However, it is challenging to generate accurate distribution patterns. One major difficulty comes from the prediction manner: since the generated density values follow the pixel-by-pixel prediction, output density maps must include spatial coherence so that they can present the smooth transition between nearest pixels. Also, the diversified scenes, e.g., irregular crowd clusters and different camera perspectives, would make the task difficult, especially for using traditional methods without deep neural networks (DNNs). The recent development of congested scene analysis relays on DNN-based methods because of the high accuracy they have achieved in semantic segmentation tasks [7, 8, 9, 10, 11] and the significant progress they have made in visual saliency [12]. The additional bonus of using DNNs comes from the enthusiastic hardware community where DNNs are rapidly investigated and implemented on GPUs [13], FPGAs [14, 15, 16], and ASICs [17]. Among them, the low-power, small-size schemes are especially suitable for deploying congested scene analysis in surveillance devices.

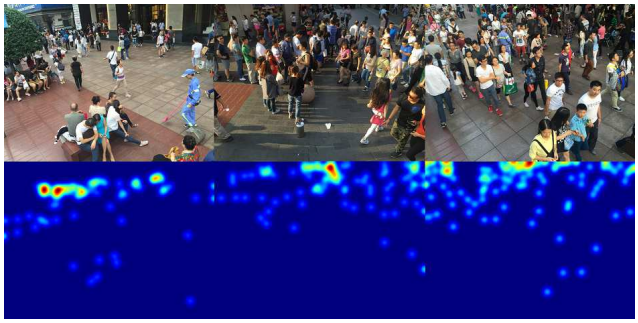

Figure 1. Pictures in first row show three images all containing 95 people in ShanghaiTech Part_B dataset [18], while having totally different spatial distributions. Pictures in second row show their density maps.

Previous works for congested scene analysis are mostly based on multi-scale architectures [4, 5, 18, 19, 20]. They have achieved high performance in this field but the designs they used also introduce two significant disadvantages when networks go deeper: large amount of training time and non-effective branch structure (e.g., multi-column CNN (MCNN) in [18]). We design an experiment to demonstrate that the MCNN does not perform better compared to a

deeper, regular network in Table 1. The main reason of using MCNN in [18] is the flexible receptive fields provided by convolutional filters with different sizes across the column. Intuitively, each column of MCNN is dedicated to a certain level of congested scene. However, the effectiveness of using MCNN may not be prominent. We present Fig. 2 to illustrate the features learned by three separated columns (representing large, medium, and small receptive fields) in MCNN and evaluate them with ShanghaiTech Part_A [18] dataset. The three curves in this figure share very similar patterns (estimated error rate) for 50 test cases with different congest densities meaning that each column in such branch structure learn nearly identical features. It performs against the original intention of the MCNN design for learning different features for each column.

In this paper, we design a deeper network called CSRNet for counting crowd and generating high-quality density maps. Unlike the latest works such as [4, 5] which use the deep CNN for ancillary, we focus on designing a CNN-based density map generator. Our model uses pure convolutional layers as the backbone to support input images with flexible resolutions. To limit the network complexity, we use the small size of convolution filters (like 3×3) in all layers. We deploy the first 10 layers from VGG-16 [21] as the front-end and dilated convolution layers as the back-end to enlarge receptive fields and extract deeper features without losing resolutions (since pooling layers are not used). By taking advantage of such innovative structure, we outperform the state-of-the-art crowd counting solutions (a MCNN based solution called CP-CNN [5]) with 7%, 47.3%, 10.0%, and 2.9% lower Mean Absolute Error (MAE) in ShanghaiTech [18] Part_A, Part_B, UCF_CC_50 [22], and WorldExpo'10 [3] datasets respectively. Also, we achieve high performance on the UCSD dataset [23] with 1.16 MAE. After extending this work to vehicle counting on TRANCOS dataset [20], we achieve 15.4% lower MAE than the current best approach, called FCN-HA [24].

The rest of the paper is structured as follows. Sec. 2 presents the previous works for crowd counting and density map generation. Sec. 3 introduces the architecture and configuration of our model while Sec. 4 presents the experimental results on several datasets. In Sec. 5, we conclude the paper.

2. Related work

Following the idea proposed by Loy *et al*. [25], the potential solutions for crowd scenes analysis can be classified into three categories: detection-based methods, regression-based methods, and density estimation-based methods. By combining the deep learning, the CNN-based solutions show even stronger ability in this task and outperform the traditional methods.

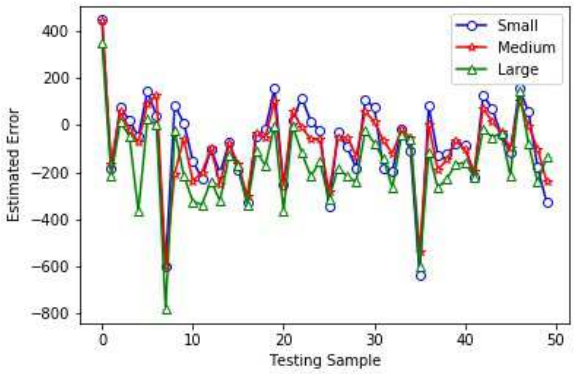

Figure 2. The estimated error of 50 samples from the testing set in ShanghaiTech Part_A [18] generated by the three pre-trained columns of MCNN. Small, Medium, Large respectively stand for the columns with small, medium or large kernels in the MCNN.

Method	Parameters	MAE	MSE
Col. 1 of MCNN	57.75k	141.2	206.8
Col. 2 of MCNN	45.99k	160.5	239.0
Col. 3 of MCNN	25.14k	153.7	230.2
MCNN Total	127.68k	110.2	185.9
A deeper CNN	83.84k	**93.0**	**142.2**

Table 1. To demonstrate that MCNN [18] may not be the best choice, we design a deeper, single-column network with fewer parameters compared to MCNN. The architecture of the proposed small network is: $CR(32, 3) - M - CR(64, 3) - M - CR(64, 3) - M - CR(32, 3) - CR(32, 3) - CR(1, 1)$. $CR(\ \ , n)$ represents the convolutional layer with m filters whose size is $n \times n$ followed by the ReLu layer. M is the max-pooling layer. Results show that the single-column version achieves higher performance on ShanghaiTech Part_A dataset [18] with the lowest MAE and Mean Squared Error (MSE)

2.1. Detection-based approaches

Most of the early researches focus on detection-based approaches using a moving-window-like detector to detect people and count their number [26]. These methods require well-trained classifiers to extract low-level features from the whole human body (like Haar wavelets [27] and HOG (histogram oriented gradients) [28]). However, they perform poorly on highly congested scenes since most of the targeted objects are obscured. To tackle this problem, researchers detect particular body parts instead of the whole body to complete crowd scenes analysis [29].

2.2. Regression-based approaches

Since detection-based approaches can not be adapted to highly congested scenes, researchers try to deploy regression-based approaches to learn the relations among extracted features from cropped image patches, and then calculate the number of particular objects. More features,

such as foreground and texture features, have been used for generating low-level information [30]. Following similar approaches, Idrees *et al*. [22] propose a model to extract features by employing Fourier analysis and SIFT (Scale-invariant feature transform) [31] interest-point based counting.

2.3. Density estimation-based approaches

When executing the regression-based solution, one critical feature, called saliency, is overlooked which causes inaccurate results in local regions. Lempitsky *et al*. [32] propose a method to solve this problem by learning a linear mapping between features in the local region and its object density maps. It integrates the information of saliency during the learning process. Since the ideal linear mapping is hard to obtain, Pham *et al*. [33] use random forest regression to learn a non-linear mapping instead of the linear one.

2.4. CNN-based approaches

Literature also focuses on the CNN-based approaches to predict the density map because of its success in classification and recognition [34, 21, 35]. In the work presented by Walach and Wolf [36], a method is demonstrated with selective sampling and layered boosting. Instead of using patch-based training, Shang *et al*. [37] try an end-to-end regression method using CNNs which takes the entire image as input and directly outputs the final crowd count. Boominathan *et al*. [19] present the first work purely using convolutional networks and dual-column architecture for generating density map. Marsden *et al*. [38] explore single-column fully convolutional networks while Sindagi *et al*. [39] propose a CNN which uses the high-level prior information to boost the density prediction performance. An improved structure is proposed by Zhang *et al*. [18] who introduce a multi-column based architecture (MCNN) for crowd counting. Similar idea is shown in Onoro and Sastre [20] where a scale-aware, multi-column counting model called Hydra CNN is presented for object density estimation. It is clear that the CNN-based solutions outperform the previous works mentioned in Sec. 2.1 to 2.3.

2.5. Limitations of the state-of-the-art approaches

Most recently, Sam *et al*. [4] propose the Switch-CNN using a density level classifier to choose different regressors for particular input patches. Sindagi *et al*. [5] present a Contextual Pyramid CNN, which uses CNN networks to estimate context at various levels for achieving lower count error and better quality density maps. These two solutions achieve the state-of-the-art performance, and both of them used multi-column based architecture (MCNN) and density level classifier. However, we observe several disadvantages in these approaches: (1) Multi-column CNNs are hard to train according to the training method described in work [18]. Such bloated network structure requires more time to train. (2) Multi-column CNNs introduce redundant structure as we mentioned in Sec. 1. Different columns seem to perform similarly without obvious differences. (3) Both solutions require density level classifier before sending pictures in the MCNN. However, the granularity of density level is hard to define in real-time congested scene analysis since the number of objects keeps changing with a large range. Also, using a fine-grained classifier means more columns need to be implemented which makes the design more complicated and causes more redundancy. (4) These works spend a large portion of parameters for density level classification to label the input regions instead of allocating parameters to the final density map generation. Since the branch structure in MCNN is not efficient, the lack of parameters for generating density map lowers the final accuracy. Taking all above disadvantages into consideration, we propose a novel approach to concentrate on encoding the deeper features in congested scenes and generating high-quality density map.

3. Proposed Solution

The fundamental idea of the proposed design is to deploy a deeper CNN for capturing high-level features with larger receptive fields and generating high-quality density maps without brutally expanding network complexity. In this section, we first introduce the architecture we proposed, and then we present the corresponding training methods.

3.1. CSRNet architecture

Following the similar idea in [19, 4, 5], we choose VGG-16 [21] as the front-end of CSRNet because of its strong transfer learning ability and its flexible architecture for easily concatenating the back-end for density map generation. In CrowdNet [19], the authors directly carve the first 13 layers from VGG-16 and add a 1×1 convolutional layer as output layer. The absence of modifications results in very weak performance. Other architectures, such as [4], uses VGG-16 as the density level classifier for labeling input images before sending them to the most suitable column of the MCNN, while the CP-CNN [5] incorporates the result of classification with the features from density map generator. In these cases, the VGG-16 performs as an ancillary without significantly boosting the final accuracy. In this paper, we first remove the classification part of VGG-16 (fully-connected layers) and build the proposed CSRNet with convolutional layers in VGG-16. The output size of this front-end network is 1/8 of the original input size. If we continue to stack more convolutional layers and pooling layers (basic components in VGG-16), output size would be further shrunken, and it is hard to generate high-quality density maps. Inspired by the works [10, 11, 40], we try to deploy dilated convolutional layers as the back-end for

extracting deeper information of saliency as well as maintaining the output resolution.

3.1.1 Dilated convolution

One of the critical components of our design is the dilated convolutional layer. A 2-D dilated convolution can be defined as follow:

$$y(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i, n + r \times j)w(i,j) \quad (1)$$

$y(m,n)$ is the output of dilated convolution from input $x(m,n)$ and a filter $w(i,j)$ with the length and the width of M and N respectively. The parameter r is the dilation rate. If $r = 1$, a dilated convolution turns into a normal convolution.

Dilated convolutional layers have been demonstrated in segmentation tasks with significant improvement of accuracy [10, 11, 40] and it is a good alternative of pooling layer. Although pooling layers (e.g., max and average pooling) are widely used for maintaining invariance and controlling overfitting, they also dramatically reduce the spatial resolution meaning the spatial information of feature map is lost. Deconvolutional layers [41, 42] can alleviate the loss of information, but the additional complexity and execution latency may not be suitable for all cases. Dilated convolution is a better choice, which uses sparse kernels (as shown in Fig. 3) to alternate the pooling and convolutional layer. This character enlarges the receptive field without increasing the number of parameters or the amount of computation (e.g., adding more convolutional layers can make larger receptive fields but introduce more operations). In dilated convolution, a small-size kernel with $k \times k$ filter is enlarged to $k + (k-1)(r-1)$ with dilated stride r. Thus it allows flexible aggregation of the multi-scale contextual information while keeping the same resolution. Examples can be found in Fig. 3 where normal convolution gets 3×3 receptive field and two dilated convolutions deliver 5×5 and 7×7 receptive fields respectively.

For maintaining the resolution of feature map, the dilated convolution shows distinct advantages compared to the scheme of using convolution + pooling + deconvolution. We pick one example for illustration in Fig. 4. The input is an image of crowds, and it is processed by two approaches separately for generating output with the same size. In the first approach, input is downsampled by a max pooling layer with factor 2, and then it is passed to a convolutional layer with a 3×3 Sobel kernel. Since the generated feature map is only 1/2 of the original input, it needs to be upsampled by the deconvolutional layer (bilinear interpolation). In the other approach, we try dilated convolution and adapt the same 3×3 Sobel kernel to a dilated kernel with a

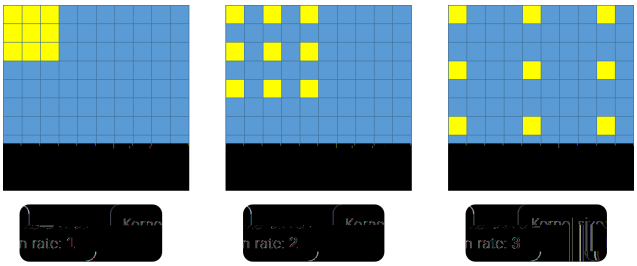

Figure 3. 3×3 convolution kernels with different dilation rate as 1, 2, and 3.

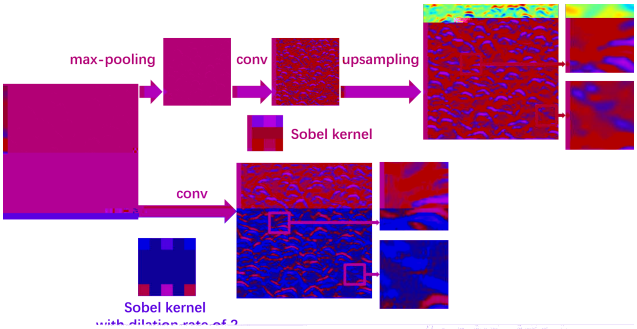

Figure 4. Comparison between dilated convolution and max-pooling, convolution, upsampling. The 3×3 Sobel kernel is used in both operations while the dilation rate is 2.

factor $= 2$ stride. The output is shared the same dimension as the input (meaning pooling and deconvolutional layers are not required). Most importantly, the output from dilated convolution contains more detailed information (referring to the portions we zoom in).

3.1.2 Network Configuration

We propose four network configurations of CSRNet in Table 3 which have the same front-end structure but different dilation rate in the back-end. Regarding the front-end, we adapt a VGG-16 network [21] (except fully-connected layers) and only use 3×3 kernels. According to [21], using more convolutional layers with small kernels is more efficient than using fewer layers with larger kernels when targeting the same size of receptive field .

By removing the fully-connected layers, we try to determine the number of layers we need to use from VGG-16. The most critical part relays on the tradeoff between accuracy and the resource overhead (including training time, memory consumption, and the number of parameters). Experiment shows a best tradeoff can be achieved when keeping the first ten layers of VGG-16 [21] with only three pooling layers instead of five to suppress the detrimental effects on output accuracy caused by the pooling operation. Since the output (density maps) of CSRNet is smaller (1/8 of input size), we choose bilinear interpolation with the factor of 8 for scaling and make sure the output shares the same

Dataset	Generating method
ShanghaiTech Part_A [18]	Geometry-adaptive kernels
UCF_CC_50 [22]	
ShanghaiTech Part_B [18]	Fixed kernel: $\sigma = 15$
TRANCOS [44]	Fixed kernel: $\sigma = 10$
The WorldExpo'10 [3]	Fixed kernel: $\sigma = 3$
The UCSD [23]	

Table 2. The ground truth generating methods for different datasets

resolution as the input image. With the same size, CSRNet generated results are comparable with the ground truth results using the PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity in Image [43]).

3.2. Training method

In this section, we provide specific details of CSRNet training. By taking advantage of the regular CNN network (without branch structures), CSRNet is easy to implement and fast to deploy.

3.2.1 Ground truth generation

Following the method of generating density maps in [18], we use the geometry-adaptive kernels to tackle the highly congested scenes. By blurring each head annotation using a Gaussian kernel (which is normalized to 1), we generate the ground truth considering the spatial distribution of all images from each dataset. The geometry-adaptive kernel is defined as:

$$F(\boldsymbol{x}) = \sum_{i=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_i) \times G_{\sigma_i}(\boldsymbol{x}),\ with\ \sigma_i = \beta \overline{d_i} \quad (2)$$

For each targeted object $\boldsymbol{x}_i$ in the ground truth δ, we use $\overline{d_i}$ to indicate the average distance of k nearest neighbors. To generate the density map, we convolve $\delta(\boldsymbol{x} - \boldsymbol{x}_i)$ with a Gaussian kernel with parameter σ_i (standard deviation), where $\boldsymbol{x}$ is the position of pixel in the image. In experiment, we follow the configuration in [18] where $\beta = 0.3$ and $k = 3$. For input with sparse crowd, we adapt the Gaussian kernel to the average head size to blur all the annotations. The setups for different datasets are shown in Table 2.

3.2.2 Data augmentation

We crop 9 patches from each image at different locations with 1/4 size of the original image. The first four patches contain four quarters of the image without overlapping while the other five patches are randomly cropped from the input image. After that, we mirror the patches so that we double the training set.

Configurations of CSRNet			
A	B	C	D
input(unfixed-resolution color image)			
front-end (fine-tuned from VGG-16)			
conv3-64-1 conv3-64-1			
max-pooling			
conv3-128-1 conv3-128-1			
max-pooling			
conv3-256-1 conv3-256-1 conv3-256-1			
max-pooling			
conv3-512-1 conv3-512-1 conv3-512-1			
back-end (four different configurations)			
conv3-512-1	conv3-512-2	conv3-512-2	conv3-512-4
conv3-512-1	conv3-512-2	conv3-512-2	conv3-512-4
conv3-512-1	conv3-512-2	conv3-512-2	conv3-512-4
conv3-256-1	conv3-256-2	conv3-256-4	conv3-256-4
conv3-128-1	conv3-128-2	conv3-128-4	conv3-128-4
conv3-64-1	conv3-64-2	conv3-64-4	conv3-64-4
conv1-1-1			

Table 3. Configuration of CSRNet. All convolutional layers use padding to maintain the previous size. The convolutional layers' parameters are denoted as "conv-(kernel size)-(number of filters)-(dilation rate)", max-pooling layers are conducted over a 2×2 pixel window with stride 2.

3.2.3 Training details

We use a straightforward way to train the CSRNet as an end-to-end structure. The first 10 convolutional layers are fine-tuned from a well-trained VGG-16 [21]. For the other layers, the initial values come from a Gaussian initialization with 0.01 standard deviation. Stochastic gradient descent (SGD) is applied with fixed learning rate at 1e-6 during training. Also, we choose the Euclidean distance to measure the difference between the ground truth and the estimated density map we generated which is similar to other works [19, 18, 4]. The loss function is given as follow:

$$L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \parallel\ Z(X_i; \Theta) - Z_i^{GT} \parallel_2^2 \quad (3)$$

where N is the size of training batch and $Z(X_i; \Theta)$ is the output generated by CSRNet with parameters shown as Θ. X_i represents the input image while Z_i^{GT} is the ground truth result of the input image X_i.

4. Experiments

We demonstrate our approach in five different public datasets [18, 3, 22, 23, 44]. Compared to the previous state-of-the-art methods [4, 5], our model is smaller, more accurate, and easier to train and deploy. In this section, the evaluation metrics are introduced, and then an ablation study of ShanghaiTech Part_A dataset is conducted to analyze the configuration of our model (shown in Table 3). Along with the ablation study, we evaluate and compare our proposed method to the previous state-of-the-art methods in all these five datasets. The implementation of our model is based on the Caffe framework [13].

4.1. Evaluation metrics

The MAE and the MSE are used for evaluation which are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \mid C_i - C_i^{GT} \mid \tag{4}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \mid C_i - C_i^{GT} \mid^2} \tag{5}$$

where N is the number of images in one test sequence and C_i^{GT} is the ground truth of counting. C_i represents the estimated count which is defined as follows:

$$C_i = \sum_{l=1}^{L}\sum_{w=1}^{W} z_{l,w} \tag{6}$$

L and W show the length and width of the density map respectively while $z_{l,w}$ is the pixel at (l, w) of the generated density map. C_i means the estimated counting number for image X_i.

We also use the PSNR and SSIM to evaluate the quality of the output density map on ShanghaiTech Part_A dataset. To calculate the PSNR and SSIM, we follow the preprocess given by [5], which includes the density map resizing (same size with the original input) with interpolation and normalization for both ground truth and predicted density map.

4.2. Ablations on ShanghaiTech Part_A

In this subsection, we perform an ablation study to analyze the four configurations of the CSRNet on ShanghaiTech Part_A dataset [18] which is a new large-scale crowd counting dataset including 482 images for congested scenes with 241,667 annotated persons. It is challenging to count from these images because of the extremely congested scenes, the varied perspective, and the unfixed resolution. These four configurations are shown in Table 3. CSRNet A is the network with all the dilation rate of 1. CSRNet B and D maintain the dilation rate of 2 and 4 in their back-end respectively while CSRNet C combines the dilated rate of 2 and 4. The number of parameters of these four models are the same as 16.26M. We intend to compare the results by using different dilation rates. After training on Shanghai Part_A dataset using the method mentioned in Sec. 3.2, we perform the evaluation metrics defined in Sec. 4.1. We try dropout [45] for preventing the potential overfitting problem but there is no significant improvement. So we do not include dropout in our model. The detailed evaluation results are shown in Table 4, where CSRNet B achieves the lowest error (the highest accuracy). Therefore, we use CSRNet B as the proposed CSRNet for the following experiments.

Architecture	MAE	MSE
CSRNet A	69.7	116.0
CSRNet B	**68.2**	**115.0**
CSRNet C	71.91	120.58
CSRNet D	75.81	120.82

Table 4. Comparison of architectures on ShanghaiTech Part_A dataset

4.3. Evaluation and comparison

4.3.1 ShanghaiTech dataset

ShanghaiTech crowd counting dataset contains 1198 annotated images with a total amount of 330,165 persons [3]. This dataset consists of two parts as Part_A containing 482 images with highly congested scenes randomly downloaded from the Internet while Part_B includes 716 images with relatively sparse crowd scenes taken from streets in Shanghai. Our method is evaluated and compared to other six recent works and results are shown in Table 5. It indicates that our method achieves the lowest MAE (the highest accuracy) in Part_A compared to other methods and we get 7% lower MAE than the state-of-the-art solution called CP-CNN. CSRNet also delivers 47.3% lower MAE in Part_B compared to the CP-CNN. To evaluate the quality of generated density map, we compare our method to the MCNN and the CP-CNN using Part_A dataset and we follow the evaluation metrics in Sec. 3.2. Samples of the test cases can be found in Fig 5. Results are shown in Table 6 which indicates CSRNet achieves the highest SSIM and PSNR. We also report the quality result of ShanghaiTech dataset in Table 11.

4.3.2 UCF_CC_50 dataset

UCF_CC_50 dataset includes 50 images with different perspective and resolutions [22]. The number of annotated persons per image ranges from 94 to 4543 with an average number of 1280. 5-fold cross-validation is performed following the standard setting in [22]. Result comparisons of

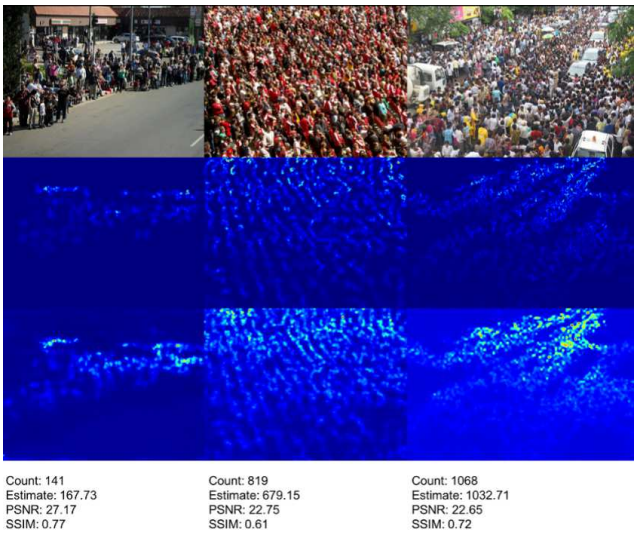

Figure 5. The first row shows the samples of the testing set in ShanghaiTech Part_A dataset. The second row shows the ground truth for each sample while the third row presents the generated density map by CSRNet.

	Part_A		Part_B	
Method	MAE	MSE	MAE	MSE
Zhang *et al*. [3]	181.8	277.7	32.0	49.8
Marsden *et al*. [38]	126.5	173.5	23.8	33.1
MCNN [18]	110.2	173.2	26.4	41.3
Cascaded-MTL [39]	101.3	152.4	20.0	31.1
Switching-CNN [4]	90.4	135.0	21.6	33.4
CP-CNN [5]	73.6	**106.4**	20.1	30.1
CSRNet (ours)	**68.2**	115.0	**10.6**	**16.0**

Table 5. Estimation errors on ShanghaiTech dataset

Method	PSNR	SSIM
MCNN [18]	21.4	0.52
CP-CNN [5]	21.72	0.72
CSRNet (ours)	**23.79**	**0.76**

Table 6. Quality of density map on ShanghaiTech Part_A dataset

MAE and MSE are listed in Table 7 while the quality of generated density map can be found in Table 11.

4.3.3 The WorldExpo'10 dataset

The WorldExpo'10 dataset [3] consists of 3980 annotated frames from 1132 video sequences captured by 108 different surveillance cameras. This dataset is divided into a training set (3380 frames) and a testing set (600 frames) from five different scenes. The region of interest (ROI) is provided for the whole dataset. Each frame and its dot maps are masked with ROI during preprocessing, and we train our model following the instructions given in Sec. 3.2. Results

Method	MAE	MSE
Idrees *et al*. [22]	419.5	541.6
Zhang *et al*. [3]	467.0	498.5
MCNN [18]	377.6	509.1
Onoro *et al*. [20] Hydra-2s	333.7	425.2
Onoro *et al*. [20] Hydra-3s	465.7	371.8
Walach *et al*. [36]	364.4	341.4
Marsden *et al*. [38]	338.6	424.5
Cascaded-MTL [39]	322.8	397.9
Switching-CNN [4]	318.1	439.2
CP-CNN [5]	295.8	**320.9**
CSRNet (ours)	**266.1**	397.5

Table 7. Estimation errors on UCF_CC_50 dataset

are shown in Table 8. The proposed CSRNet delivers the best accuracy in 4 out of 5 scenes and it achieves the best accuracy on average.

Method	Sce.1	Sce.2	Sce.3	Sce.4	Sce.5	Avg.
Chen *et al*. [46]	2.1	55.9	9.6	11.3	3.4	16.5
Zhang *et al*. [3]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [18]	3.4	20.6	12.9	13.0	8.1	11.6
Shang *et al*. [37]	7.8	15.4	14.9	11.8	5.8	11.7
Switching-CNN [4]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [5]	2.9	14.7	10.5	**10.4**	5.8	8.86
CSRNet (ours)	**2.9**	**11.5**	**8.6**	16.6	**3.4**	**8.6**

Table 8. Estimated errors on the WorldExpo'10 dataset

4.3.4 The UCSD dataset

The UCSD dataset [23] has 2000 frames captured by surveillance cameras. These scenes contain sparse crowd varying from 11 to 46 persons per image. The region of interest (ROI) is also provided. Because the resolution of each frame is fixed and small (238×158), it is difficult to generate a high-quality density map after frequent pooling operations. So we preprocess the frames by using bilinear interpolation to resize them into 952×632. Among the 2000 frames, we use frames 601 through 1400 as training set and the rest of them as testing set according to [23]. Before blurring the annotation as we mentioned in Sec. 3.2, all the frames and the corresponding dot maps are masked with ROI. The accuracy of running UCSD dataset is shown in Table 9 and we outperform most of the previous methods except MCNN in the MAE category. Results indicate that our method can perform not only counting tasks for extremely dense crowds but also tasks for relative sparse scenes. Also, we provides the quality of generated density map in Table 11.

4.3.5 TRANCOS dataset

Beyond the crowd counting, we setup an experiment on the TRANCOS dataset [44] for vehicle counting to

Method	MAE	MSE
Zhang *et al*. [3]	1.60	3.31
CCNN [20] CCNN	1.51	-
Switching-CNN [4]	1.62	2.10
FCN-rLSTM [24]	1.54	3.02
CSRNet (ours)	1.16	1.47
MCNN [18]	**1.07**	**1.35**

Table 9. Estimation errors on the UCSD dataset

demonstrate the robustness and generalization of our approach. TRANCOS is a public traffic dataset containing 1244 images of different congested traffic scenes captured by surveillance cameras with 46796 annotated vehicles. Also, the region of interest (ROI) is provided for the evaluation. The perspectives of images are not fixed and the images are collected from very different scenarios. The Grid Average Mean Absolute Error (GAME) [44] is used for evaluation in this test. The GAME is defined as follow:

$$GAME(L) = \frac{1}{N}\sum_{n=1}^{N}(\sum_{l=1}^{4^L}\left|D^l_{I_n} - D^l_{I^{gt}_n}\right|) \qquad (7)$$

where N is the number of images in testing set, and $D^l_{I_n}$ is the estimated result of the input image n within region l. $D^l_{I^{gt}_n}$ is the corresponding ground truth result. For a specific level L, the GAME(L) subdivides the image using a grid of 4^L non-overlapping regions which cover the full image, and the error is computed as the sum of the MAE in each of these regions. When $L = 0$, the GAME is equivalent to the MAE metric.

We compare our approach with the previous state-of-the-art methods [47, 32, 20, 24]. The method in [20] uses the MCNN-like network to generate the density map while the model in [24] deploys a combination of fully convolutional neural networks (FCN) and a long short-term memory network (LSTM). Results are shown in Table 10 with three examples shown in Fig. 6. Our model achieves a significant improvement on four different GAME metrics. Compared to the result from [20], CSRNet delivers 67.7% lower GAME(0), 60.1% lower GAME(1), 48.7% lower GAME(2), and 22.2% lower GAME(3), which is the best solution. We also present the quality of generated density map in Table 11.

Method	GAME 0	GAME 1	GAME 2	GAME 3
Fiaschi *et al*. [47]	17.77	20.14	23.65	25.99
Lempitsky *et al*. [32]	13.76	16.72	20.72	24.36
Hydra-3s [20]	10.99	13.75	16.69	19.32
FCN-HA [24]	4.21	-	-	-
CSRNet (Ours)	**3.56**	**5.49**	**8.57**	**15.04**

Table 10. GAME on the TRANCOS dataset

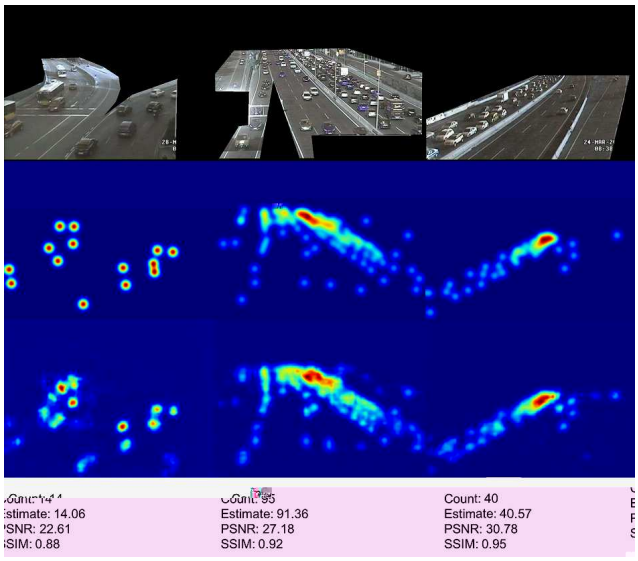

Figure 6. The first row shows samples of the testing set in TRANCOS [44] dataset with ROI. The second row shows the ground truth for each sample. The third row shows the generated density map by CSRNet.

Dataset	PSNR	SSIM
ShanghaiTech Part_A [18]	23.79	0.76
ShanghaiTech Part_B [18]	27.02	0.89
UCF_CC_50 [22]	18.76	0.52
The WorldExpo'10 [3]	26.94	0.92
The UCSD [23]	20.02	0.86
TRANCOS [44]	27.10	0.93

Table 11. The quality of density maps generated by CSRNet in 5 datasets

5. Conclusion

In this paper, we proposed a novel architecture called CSRNet for crowd counting and high-quality density map generation with an easy-trained end-to-end approach. We used the dilated convolutional layers to aggregate the multi-scale contextual information in the congested scenes. By taking advantage of the dilated convolutional layers, CSRNet can expand the receptive field without losing resolution. We demonstrated our model in four crowd counting datasets with the state-of-the-art performance. We also extended our model to vehicle counting task and our model achieved the best accuracy as well.

6. Acknowledgement

This work was supported by the IBM-Illinois Center for Cognitive Computing System Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network.

References

[1] Beibei Zhan, Dorothy N Monekosso, Paolo Remagnino, Sergio A Velastin, and Li-Qun Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008.

[2] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan. Crowded scene analysis: A survey. *IEEE transactions on circuits and systems for video technology*, 25(3):367–386, 2015.

[3] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015.

[4] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 6, 2017.

[5] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2017.

[6] Cong Zhang, Kai Kang, Hongsheng Li, Xiaogang Wang, Rong Xie, and Xiaokang Yang. Data-driven crowd understanding: a baseline for a large-scale crowd dataset. *IEEE Transactions on Multimedia*, 18(6):1048–1061, 2016.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[8] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320, 2017.

[9] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, 2017.

[10] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[11] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, 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*, PP(99):1–1, 2017.

[12] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, 2016.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[14] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, Yu Wang, and Huazhong Yang. Going deeper with embedded FPGA platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '16, pages 26–35, New York, NY, USA, 2016. ACM.

[15] Xiaofan Zhang, Xinheng Liu, Anand Ramachandran, Chuanhao Zhuge, Shibin Tang, Peng Ouyang, Zuofu Cheng, Kyle Rupnow, and Deming Chen. High-performance video content recognition with long-term recurrent convolutional network for FPGA. In *Field Programmable Logic and Applications (FPL), 2017 27th International Conference on*, pages 1–4. IEEE, 2017.

[16] Xiaofan Zhang, Anand Ramachandran, Chuanhao Zhuge, Di He, Wei Zuo, Zuofu Cheng, Kyle Rupnow, and Deming Chen. Machine learning on FPGAs to face the IoT revolution. In *Computer-Aided Design (ICCAD), 2017 IEEE/ACM International Conference on*, pages 819–826. IEEE, 2017.

[17] Renzo Andri, Lukas Cavigelli, Davide Rossi, and Luca Benini. Yodann: An ultra-low power convolutional neural network accelerator based on binary weights. In *VLSI (ISVLSI), 2016 IEEE Computer Society Annual Symposium on*, pages 236–241. IEEE, 2016.

[18] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016.

[19] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: a deep convolutional network for dense crowd counting. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 640–644. ACM, 2016.

[20] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[23] A. B. Chan, Zhang-Sheng John Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008.

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

[25] Chen Change Loy, Ke Chen, Shaogang Gong, and Tao Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*, pages 347–382. Springer, 2013.

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

[27] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[28] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[29] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[30] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 545–551. IEEE, 2009.

[31] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[32] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, pages 1324–1332, 2010.

[33] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 3253–3261. IEEE, 2015.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[35] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, July 2017.

[36] Elad Walach and Lior Wolf. Learning to count with CNN boosting. In *European Conference on Computer Vision*, pages 660–676. Springer, 2016.

[37] Chong Shang, Haizhou Ai, and Bo Bai. End-to-end crowd counting via joint learning local and global count. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1215–1219. IEEE, 2016.

[38] Mark Marsden, Kevin McGuiness, Suzanne Little, and Noel E O'Connor. Fully convolutional crowd counting on highly congested scenes. *arXiv preprint arXiv:1612.00220*, 2016.

[39] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017.

[40] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[41] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[42] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[44] Roberto Lpez-Sastre Saturnino Maldonado Bascn Ricardo Guerrero-Gmez-Olmedo, Beatriz Torre-Jimnez and Daniel Ooro-Rubio. Extremely overlapping vehicle counting. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2015.

[45] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[46] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy. Cumulative attribute space for age and crowd density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2467–2474, 2013.

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