Perspective-Guided Convolution Networks for Crowd Counting

Zhaoyi Yan[1][†], Yuchen Yuan[2], Wangmeng Zuo[1,3][*], Xiao Tan[2], Yezhen Wang[1], Shilei Wen[2], Errui Ding[2]

[1]Harbin Institute of Technology, [2]Department of Computer Vision Technology (VIS), Baidu Inc., [3]Peng Cheng Laboratory, Shenzhen

yanzhaoyi@outlook.com, wmzuo@hit.edu.cn

{tanxchong, yezhen.wang0305}@gmail.com

{yuanyuchen02, wenshilei, dingerrui}@baidu.com

Abstract

In this paper, we propose a novel perspective-guided convolution (PGC) for convolutional neural network (CNN) based crowd counting (i.e. PGCNet), which aims to overcome the dramatic intra-scene scale variations of people due to the perspective effect. While most state-of-the-arts adopt multi-scale or multi-column architectures to address such issue, they generally fail in modeling continuous scale variations since only discrete representative scales are considered. PGCNet, on the other hand, utilizes perspective information to guide the spatially variant smoothing of feature maps before feeding them to the successive convolutions. An effective perspective estimation branch is also introduced to PGCNet, which can be trained in either supervised setting or weakly-supervised setting when the branch has been pre-trained. Our PGCNet is single-column with moderate increase in computation, and extensive experimental results on four benchmark datasets show the improvements of our method against the state-of-the-arts. Additionally, we also introduce Crowd Surveillance, a large scale dataset for crowd counting that contains 13,000+ high-resolution images with challenging scenarios. Code and pre-trained models are available at https://github.com/Zhaoyi-Yan/PGCNet.

1. Introduction

The growth of global population and urbanization has been consistently promoting the frequency of crowd gathering. In such scenarios, stampedes and crushes can be life threatening and should always be prevented. Congested scene analysis and understanding is thus essential to the management, control, and security guarding of crowd gathering in cities. Among the developments in congested scene analysis, crowd counting [28, 10] is one of the fundamental
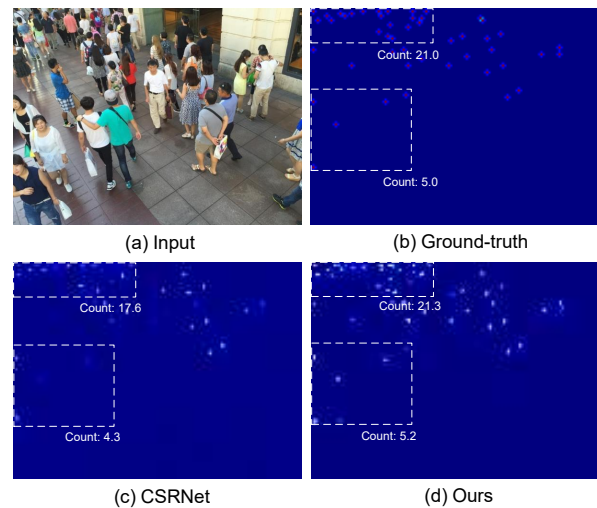

Figure 1: Density map estimations by CSRNet [11] and our PGC-Net. The MAE of our PGCNet is 2.8, much lower than that of CSRNet (7.7). It is observed that PGCNet has consistent better performances at either smaller or larger scales among the marked regions.

tasks, and recently has drawn considerable attention from the computer vision community.

Single image based crowd counting remains an active but challenging topic due to the complex distribution of people, non-uniform illumination, inter- and intra-scene scale variations, cluttering and occlusions, etc. Existing crowd counting methods can be broadly classified into three categories, *i.e.* detection-based [22, 14], regression-based [7, 27], and CNN-based methods [11, 15]. Among them, CNN-based methods have been studied in depth in the past few years, and have achieved superior performances in terms of accuracy and robustness.

However, the dramatic intra-scene scale variations of people due to the perspective effect forms a major challenge. Existing methods [6, 19, 23, 26, 29, 30] usually adopt multi-scale or multi-column architectures to fuse the

features of different scales. Yet, they suffer from several limitations. Firstly, the multi-column architectures (*e.g.* MCNN [30]) are ineffective to train. As shown in [11], MCNN cannot even compete with a deeper CNN due to the high correlation of the features learned by different columns. Secondly, they only consider discrete scales, which is limited when addressing continuous scale variations in practical scenarios. Thirdly, the computational cost increases linearly with the growth of columns or scales.

In [11], a deeper CNN (CSRNet) with dilated convolutions has achieved state-of-the-art performance. Nevertheless, it still delivers fixed receptive field for different scales of people, thereby remaining vulnerable to the highly variant intra-scene scales. It is seen in Fig. 1(c) that CSRNet performs well at intermediate scales, but behaves relatively poor at smaller or larger scales. We thus take a step forward to propose the perspective-guided convolutional network (PGCNet), which is a single-column CNN that aims to tackle the continuous scale variation issue with perspective information considered.

The perspective information encodes the distance between camera and a scene, which serves as a reasonable scale estimation of people. We thus adopt it to allocate spatially variant receptive fields, thereby conducting scale adaptive density map estimation. To this end, we propose a novel perspective-guided convolution (PGC), in which the perspective information functions to guide the spatially variant smoothing of feature maps before taking them to the successive convolutions. As a result, larger (or smaller) Gaussian kernels for feature smoothing are adopted for people at larger (or smaller) scales. After such spatially variant feature smoothing, the conventional spatially invariant convolution is appended, which forms a PGC block. It is worth noting that PGC serves as an insertable module to existing architectures, and our PGCNet is formulated by stacking multiple PGC blocks upon a CNN backbone.

However, off-the-rack perspective annotations are seldom available for existing datasets. We hence introduce a perspective estimation branch to PGCNet, which can be learned either in supervised or weakly-supervised setting when the branch has been pre-trained.

Experimental results on benchmark datasets against the state-of-the-arts show the favorable performance of our proposed PGCNet in handling intra-scene scale variations. In addition, we also introduce Crowd Surveillance, a large scale dataset for crowd counting that contains 10,000+ high-resolution images with complicated backgrounds and varied crowd counts. This dataset will be released as a new benchmark to facilitate crowd counting researches. To sum up, the main contribution of this work includes:

(1) The PGC, as an insertable module, is proposed to handle the intra-scene scale variations of crowd counting;

(2) A perspective estimation branch is introduced, which can be trained with or without perspective annotations;

(3) An end-to-end trainable PGCNet is formulated with (1) and (2);

(4) A new large scale dataset is introduced;

(5) State-of-the-art performance is achieved by PGC-Net on four benchmark datasets, *e.g.* 57.0 MAE on ShanghaiTech Part A and 8.8 MAE on ShanghaiTech Part B.

2. Related Work

Crowd counting methods can be roughly categorized into three subsets, *i.e.* detection-based, regression-based and CNN-based methods. In this paper, we only review CNN-based methods, which are the most related to our method. Besides, the exploration on perspective normalization in crowd counting is also surveyed.

2.1. CNN-based Methods

Benefited from the great success of CNN, many CNN-based works of crowd counting have been proposed in recent years. These methods usually focus on typical techniques, including multi-scale [30, 23, 26, 16, 17, 24], context [26], multi-task [29, 6, 15], and others [12, 13, 21]. Recently, more methods have been proposed for handling the scale variation issue. For instance, Zhang *et al*. [30] suggest a multi-column architecture (MCNN) that combines features with different sizes of receptive fields. In Switching-CNN [23], one of the three regressors is assigned for an input image in refer to its specific crowd density. CP-CNN [26] incorporates MCNN with local and global contexts. SANet [2] employs scale aggregation modules for multi-scale representation. And instead of multi-column architecture, CSRNet [11] enlarges receptive fields by stacking dilated convolutions.

Our proposed method differs from existing CNN-based method in two aspects: firstly, our method is able to handle continuous scale variations of each single pedestrian in the image, instead of simply fusing the features of different scales; secondly, the perspective information is taken into account as a vital estimator of pedestrian scales.

2.2. Perspective Normalization

Perspective is originally adopted in the normalization of extracted features from foreground objects in [3]. Later, Lempitsky *et al*. [9] attempt to deal with perspective distortion by optimizing the loss on the MESA-distance. Among the CNN-based methods, the perspective information is usually exploited as part of the pre-processing to generate the density maps [29, 30, 23, 6], but is seldom directly encoded into the network architecture. One of the most relevant work to our method is PACNN [17]. However, PACNN is still based on the multi-column architecture with discrete scales. In PACNN, two density maps are estimated from

two columns based on VGG-16 [25] backbone. The two predicted density maps are assigned weights generated by the perspective map and later are combined together as the final estimation. However, we argue that a better choice would be an explicit perspective normalization during the training of the network itself, which is able to tackle the continuous scale variation issue, as illustrated in Sec. 3.1.

3. Proposed Method

In this section, we first describe the principles of PGC, then introduce a perspective estimation branch for perspective map estimation, and finally provide the network architecture and learning objective of the complete PGCNet.

3.1. Perspective-Guided Convolution

To handle the intra-scene scale variations mentioned in Sec. 1, it is desirable to use a larger (or smaller) receptive field for people at larger (or smaller) scales. Let $\mathbf{x}$ and $\mathbf{p}$ be the feature map and perspective map of an image, respectively. Without loss of generality, $\mathbf{p}$ is downsampled to the same size $h \times w$ with $\mathbf{x}$. A straightforward way is to apply spatially variant convolutions by assigning scale-aware kernel sizes based on the corresponding perspective values, *i.e.* kernels with different sizes may be used at different locations of a feature map. However, several drawbacks hinder such solution: (1) the discrete kernel sizes might be incompatible with the continuous perspective values, (2) arbitrary spatially variant convolution is hard to implement and optimize, and (3) additional constraints are required to effectively enforce the consistency among different scales [24].

To address the issues above, we propose perspective-guided convolution (PGC), which functions in a two-stage scheme: spatially variant Gaussian filtering (which smooths the feature maps in a spatially variant way) and spatially invariant convolution (*i.e.* the conventional convolution). To begin with, the perspective map $\mathbf{p}$ is normalized as,

$$\widetilde{\mathbf{p}} = \zeta\left(\mathbf{p}\right) = \frac{1}{1 + e^{-\alpha(\mathbf{p}-\beta)}} \quad (1)$$

where $\zeta\left(x\right)$ is a sigmoid-like function, and α and β are two parameters learned during training. We then define the blur map as,

$$\boldsymbol{\sigma} = \max(a(\widetilde{\mathbf{p}} - p_0), 0), \quad (2)$$

where a and p_0 are another two trainable parameters. Thus, in the spatially variant Gaussian filtering, the smoothing result $\widetilde{x}_{i,j}$ at (i, j) can be obtained by

$$\widetilde{x}_{i,j} = \sum_k \sum_l x_{k,l} G_{\sigma_{i,j}}(i, j, k, l), \quad (3)$$

where $G_{\sigma_{i,j}}(i, j, k, l)$ is a Gaussian kernel with standard deviation $\sigma_{i,j}$ centered at (i, j),

$$G_{\sigma_{i,j}}(i, j, k, l) = \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \exp\left(-\frac{((k-i)^2 + (l-j)^2)}{2\sigma_{i,j}^2}\right). \quad (4)$$

The perspective-guided convolution can then be defined as,

$$\mathbf{y} = \mathbf{W}^T \widetilde{\mathbf{x}}, \quad (5)$$

where $\mathbf{W}$ denotes the spatially invariant convolution kernel.

However, Eqn. (3) is computationally heavy. We thus present an efficient version for approximation. First, we sample N candidate Gaussian filters with size $K \times K$ and standard deviation in the pre-defined range $[c, d]$. After that, principal component analysis (PCA) is performed on the candidates to obtain the eigenvectors $\{G_q\}_{q=1}^C$ corresponding to the C non-zero eigenvalues. For each G_q, we define the coefficient map $\mathbf{u}_q$ with its (i, j)-th element as,

$$\mathbf{u}_q(i, j) = \langle G_q, G_{\sigma_{i,j}} \rangle, \quad (6)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. With $\mathbf{u}_q$, spatially variant Gaussian smoothing can then be approximated as,

$$\widetilde{\mathbf{x}} = \sum_{q=1}^C \mathbf{u}_q \circ (\mathbf{x} * G_q), \quad (7)$$

where $\circ$ denotes entry-wise product, and $*$ is convolution operation. Due to the fact that Gaussian filter is isotropic, C is generally smaller than $(K + 1)/2$. When N is large, Eqn. (7) is guaranteed to be a good and efficient approximation. The efficiency is given more descriptions here. Take the feature of size $C \times H \times W$ as an example, Eqn. (3) has to perform pixel-wise matrix multiplication (MM) for each $x_{k,l}$. In comparison, Eqn. (6) only need $C \times i$ times MM, by the observation that $G_{\sigma_{i,j}}$ shares the same value for each j. Finally, Eqn. (7) only takes about $1/10$ the time compared to Eqn. (6), making the acceleration up to $\sim$20 times for most cases. For the degree of approximation, it is measured by the energy preserved. Please refer to Section. 5.1 for more details.

There are a few notes about PGC that are worthwhile to mention. With the spatially variant Gaussian smoothing, PGC can adaptively employ a larger (or smaller) receptive field for people at larger (or smaller) scale, which handles the continuous intra-scene scale variations flexibly. On the other hand, to effectively enforce the consistency for handling different scales, the conventional spatially invariant convolution is appended in the second stage of PGC, as indicated in Eqn. (5). Furthermore, it is seen from Eqns. (5) and (7) that the complete set of parameters of PGC can be trained end-to-end.

3.2. Perspective Estimation

Since perspective map annotations are seldom available, we introduce a perspective estimation branch to learn the perspective map of an image. Ideally, with the annotated density map for crowd counting alone, the entire model (including the perspective estimation branch) can be trained
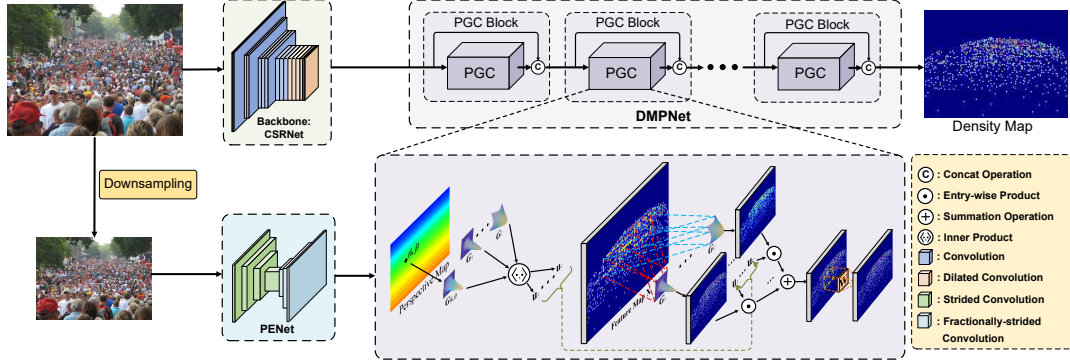
Figure 2: The architecture of the proposed PGCNet. The DMPNet takes the backbone features and the estimated perspective map from PENet to generate the final density map estimation.

end-to-end. However, such strategy ignores the internal structure of the perspective map and may result in very poor results. Motivated by [18], we suggest a three-phase procedure to train an auto-encoder.

In the first phase, we use perspective maps from WorldExpo'10 [29] to train an auto-encoder $D_p(E_p(\mathbf{p};\Theta_p^E);\Theta_p^D)$, which takes a perspective map $\mathbf{p}$ as input and reconstructs it. Θ_p^E and Θ_p^D denote the model parameters of the encoder E_p and decoder D_p, respectively. The objective is to minimize the ℓ_2 reconstruction loss,

$$\mathcal{L}_{p2p} = \frac{1}{2N}\sum_{i=1}^{N}\|D_p(E_p(\mathbf{p}_i;\Theta_p^E);\Theta_p^D) - \mathbf{p}_i\|_2^2, \quad (8)$$

where N denotes the number of perspective maps. After training, the latent code $E_p(\mathbf{p};\Theta_p^E)$ can encode the internal structure and contextual relationships of $\mathbf{p}$, while the decoder D_p can accurately and robustly recover a high quality perspective map from the latent code.

In the second phase, we use the image and perspective map pairs in WorldExpo'10 to learn another auto-encoder $D_p(E_I(\mathbf{I};\Theta_I^E);\Theta_p^D)$, which takes an image I as input to predict its perspective map. Here, we adopt the decoder parameters Θ_p^D from the first phase, and only train the encoder parameters Θ_I^E by minimizing the following loss,

$$\mathcal{L}_{I2p} = \frac{1}{2N}\sum_{i=1}^{N}\|D_p(E_I(I_i;\Theta_I^E);\Theta_p^D) - \mathbf{p}_i\|_2^2. \quad (9)$$

In the third phase, we further train the encoder with another training set. The encoder is fine-tuned with the loss of the density map if perspective maps are unavailable, and can be optimized by both $\mathcal{L}_{I2p}$ and density map loss if perspective maps are available. Benefited from the robustness of the decoder, even if the encoder is not well trained, the decoder can still recover a reasonable perspective map.

Based on the description above, for a new set of training data, our perspective estimation branch can be trained even without corresponding perspective annotations.

3.3. Network Architecture

Fig. 2 illustrates the architecture of our PGCNet, which is comprised of three subnetworks, *i.e.* backbone, perspective estimation network (PENet), and density map predictor network (DMPNet). We adopt CSRNet [11] as our backbone, where the last convolution (*i.e. conv1-1-1*) is removed. The PENet uses the encoder-decoder structure in Sec. 3.2, which will be described in detail in supplementary material. As for the DMPNet, in each PGC module, a dilated convolution with factor of 2 is exploited after the spatially variant Gaussian smoothing. The PGC block is then constructed by concatenating the features before / after the PGC module. Finally, the outputs from the backbone and PENet are fed to the DMPNet, which stacks five PGC blocks for density map estimation.

4. The Crowd Surveillance Dataset

Limited by the annotation difficulty, most public datasets for crowd counting are of relatively small size (as shown in Table 1). Although larger datasets (*i.e.* WorldExpo'10 [29]) have been proposed, they are nevertheless of low resolution and image quality. We hence introduce the Crowd Surveillance dataset*, which contains 13,945 high-resolution images (386,513 marked people). This means that Crowd Surveillance is nearly $3\times$ larger than the combination of all the other four datasets in Table 1, leading to the largest dataset with the highest average resolution for crowd counting at present. Besides, we also provide regions of interest (ROI) annotation for each image to mask out the regions that are too blurry or ambiguous for training / testing.

We build our dataset by both online crawling with search engines and real-life surveillance video acquisition from

---
*https://ai.baidu.com/broad/subordinate?dataset=crowd_surv

Dataset	#Train	#Test	Avg. Density	Avg. Resolution
ShanghaiTech A	300	182	500	868×589
ShanghaiTech B	400	316	123	1,024×768
WorldExpo'10	3,380	600	56	720×576
UCF_CC_50	-	50	1,279	902×653
Crowd Surveillance	**10,880**	**3,065**	**35**	**1,342×840**

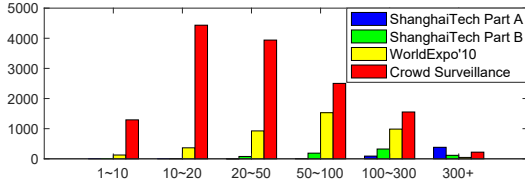Table 1: Statistics of different crowd counting datasets.

Figure 3: Statistical histogram of crowd counts on different datasets. It is seen that our dataset exhibits high data volume and crowd count variance.

Figure 4: Qualitative comparison of crowd counting datasets. Our dataset is seen to have more challenging scenarios and higher resolutions than existing datasets.

cooperative partners with necessary permissions. Fig. 3 provides the statistical histogram of crowd count on different datasets, among which Crowd Surveillance exhibits remarkable high data volume and crowd count variances. Fig. 4 shows the qualitative comparison between our dataset and two most relevant benchmarks, ShanghaiTech Part A [30] and WorldExpo'10 [29]. From Fig. 4 and Table 1, it is seen that although our dataset is less crowded in average density, it provides the highest average resolution (which ensures the image quality), and covers more challenging scenarios with complicated backgrounds and varying crowd count, which significantly increases the difficulty of crowd density estimation. It is also worth noting that although datasets with extremely high crowd densities (such as UCF-QNRF [8]) have been proposed, they are less suitable for real-world surveillance scenarios, which usually have moderate-to-low crowd density. Our dataset, on the

other hand, fits these practical applications well.

5. Experimental Results

In this section, we first present our implementation details, and then compare the proposed PGCNet with the state-of-the-arts on four public datasets, namely ShanghaiTech [30], WorldExpo'10 [29], UCF_CC_50 [7] and our proposed Crowd Surveillance. Extensive ablation study is then conducted to reveal the contribution of each component in PGCNet. We adopt mean absolute error (MAE) and mean square error (MSE) as metrics for evaluating crowd counting and perspective estimation.

5.1. Implementation Details

Denote by Θ the model parameters of the full PGCNet $\Phi(I; \Theta)$. Given the training data $\{(I_i, Y_i)\}$, the network can be trained by minimizing the following objective function,

$$L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \|\Phi(I; \Theta) - Y_i\|_2^2, \qquad (10)$$

where Y_i is the ground-truth density map of I_i. If perspective map annotations are also available, we can further incorporate $\mathcal{L}_{I2p}$ in Eqn. (9) for better training of the PENet.

To train PGCNet, we adopt stochastic gradient descent (SGD) with fixed learning rate 10^{-7} and weight decay 5×10^{-4}. The momentum is set to 0.95 and the batch size is set to 1. PGCNet takes a whole image as input during both training and testing. Suppose there is a dot annotation of people head at location (x_i, y_i) represented by a delta function $\delta(x - x_i)\delta(y - y_i)$, the ground-truth density map Y is obtained by convolving each annotation point with a normalized Gaussian kernel G_σ: $Y = \sum_i^S \delta(x - x_i)\delta(y - y_i) * G_\sigma$, where S is the total number of dot annotations in the image, and σ is set to fixed value 0.5.

To establish the baseline, we reimplement CSRNet [11] on Pytorch [20] (denoted as CSRNet*) with the four datasets adopted, which is expected to deliver comparable performances against [11]. The weights of the PGC modules are initialized with Gaussian distribution of zero mean and 0.01 standard deviation, while the other layers are initialized with the corresponding pre-trained weights from CSRNet*. Random flipping is adopted for data augmentation. For ShanghaiTech and WorldExpo'10, since their ground-truth perspective maps are available, they are used as the guidance of PGC directly, and our model is trained for 300 epochs without PENet. For the datasets without perspective annotations, the PENet is pre-trained for the first two phases of Sec. 3.2, both 500 epochs; the backbone, PENet and DMPNet are then trained together for another 300 epochs, as the third phase of Sec. 3.2.

The range of σ we sample is $[1/4, K/4]$ with $P\{\|X - \mu\| < 2\sigma\} \approx 0.95$. We pre-define a group of Gaussian candidates G_σ of size $K \times K$ with σ ranging in $[1/4, K/4]$ on

the step s. K is set to 7 by default, unless explicitly stated; and s is set to 0.05 since no significant gain is observed with denser steps. The total number of Gaussian candidates is $N = 30$ with σ sampling in $[1/4, 7/4]$ on step $s = 0.05$. PCA is then applied on G_σ to get the $(K+1)/2$ eigenvectors G_q in Eqn. (6). In this case, the original matrix is $30 \times 7 \times 7$, which will be reshaped to 30×49 and applied PCA to get the approximate matrix 4×49. The energy preserved is 91.2%, which is the ratio between the sum of singular values of approximated matrix against those of the original matrix. α and β are initialized by normalizing the $\mathbf{p}$ in Eqn. (1) to $(0, 1)$, while p_0 and a are empirically set to 0 and 1. All the parameters above are differentiable and trainable. As a tradeoff between efficiency and performance, our results are achieved by stacking five PGC blocks for all datasets. It takes about 2 days to train the network on WolrdExpo'10 with an NVIDIA Tesla P40 GPU.

5.2. Evaluations and Comparisons

Four datasets are adopted in our experiments, including ShanghaiTech [30], WorldExpo'10 [29], UCF_CC_50 [7] and the proposed Crowd Surveillance. For the later two that do not have perspective map annotations, we denote *Ours A* as directly adopting the estimated perspective maps (based on the PENet trained on perspective annotations from ShanghaiTech A) as *ground-truth* and feed them to the training of PGC; while *Ours B* as the end-to-end training without perspective map annotations described in Sec. 3.2, where the backbone, PENet, and DMPNet are jointly trained without any density map annotation. For the estimated perspective maps of both *Ours A* and *Ours B*, we use the mean of each line to replace the values of the whole line, which forms the *final* estimated perspective map. For *Ours A*, by following Sec. 3.2 with ShanghaiTech A, we get 0.020 MAE and 0.031 MSE for the first phase of perspective map estimation, and 0.101 MAE and 0.142 MSE for the second phase. Visualization of perspective estimation and more details of training of the PENet will be illustrated in our supplementary materials.

ShanghaiTech contains 1,198 images with a total of 330,165 annotated people. The dataset is split into Part A and Part B, with 482 and 716 images respectively.

We adopt the ground-truth perspective maps provided by [17] as the guidance of DMPNet. The results are listed in Table 2. It is seen that our method and SANet [2] dominate the top ranks, where ours achieves the best result on Part A w.r.t. both MAE and MSE with significant margins, and is only slightly surpassed by SANet on Part B. Our method also exhibits significant performance gain over the baseline CSRNet*. Besides, some test cases can be found in Fig. 5, clearly indicating that the superiority of PGCNet over CSRNet in estimating a better density map.

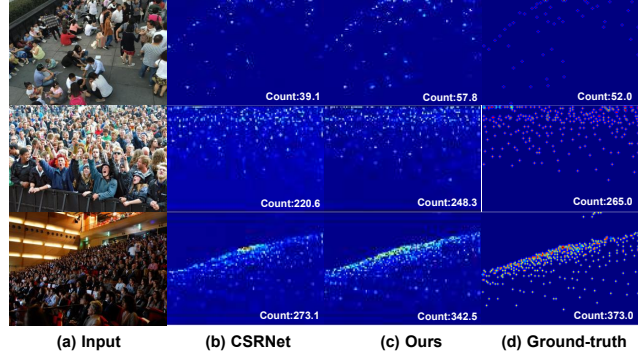**WorldExpo'10** contains 3,980 images from the 2010

(a) Input   (b) CSRNet   (c) Ours   (d) Ground-truth

Figure 5: Density maps estimated by CSRNet [11] and Ours.

Method	Part A		Part B	
	MAE	MSE	MAE	MSE
Zhang *et al*. [29]	181.8	277.7	32.0	49.8
MCNN [30]	110.2	173.2	26.4	41.3
Cascaded-MTL [1]	101.3	152.4	20.0	31.1
Switching-CNN [23]	90.4	135.0	21.6	33.4
CP-CNN [26]	73.6	106.4	20.1	30.1
PACNN [17]	84.5	132.5	14.2	24.1
DecideNet [12]	-	-	20.8	29.4
SANet [2]	67.0	104.5	**8.4**	**13.6**
CSRNet [11]	68.2	115.0	10.6	16.0
CSRNet*	67.5	103.1	10.7	16.4
Ours	**57.0**	**86.0**	8.8	13.7

Table 2: Comparisons on ShanghaiTech dataset [30].

Shanghai WorldExpo. The training set contains 3,380 images, while the test set is divided into five different scenes with 120 images each. ROIs are provided to indicate the target regions for training / testing. Following [11], each image and its ground-truth density map are masked with the ROI in preprocessing. We use the official ground-truth perspective map to guide the processing of PGC. The results are shown in Table 3, where our method achieves the best 8.1 average MAE against other methods.

Method	Sce.1	Sce.2	Sce.3	Sce.4	Sce.5	Avg.
Zhang *et al*. [29]	9.6	14.1	14.3	22.2	3.7	12.9
MCNN [30]	3.4	20.6	12.9	13.0	8.1	11.6
Switching-CNN [23]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [26]	2.9	14.7	10.5	10.4	5.8	8.9
PACNN [17]	2.6	15.4	10.6	10.0	4.8	8.7
DecideNet [12]	2.0	13.1	8.9	17.4	4.8	9.2
SANet [2]	2.6	13.2	9.0	13.3	3.0	8.2
CSRNet [11]	2.9	11.5	8.6	16.6	3.4	8.6
CSRNet*	2.4	15.1	7.9	15.6	2.7	8.7
Ours	2.5	12.7	8.4	13.7	3.2	**8.1**

Table 3: Comparisons on WorldExpo'10 [29] dataset.

UCF_CC_50 contains 50 images of diverse scenes. The head count per image varies drastically (from 94 to 4,543). Following [7], we split the dataset into five subsets and perform a 5-fold cross-validation. Since the perspective map

annotation is unavailable, we conduct the two comparison experiments *Ours A* and *Ours B* described at the beginning of this section. The results are shown in Table 4, where our method achieves the optimal performance with large margins. Compared with the baseline CSRNet*, *Ours A* achieves significant gain on both MAE and MSE.

We further note that *Ours B* achieves 244.6 MAE, with another 14.8 gain over *Ours A*, which shows the feasibility of our end-to-end training strategy described in Sec. 3.2.

Method	MAE	MSE
Idrees *et al*. [7]	419.5	541.6
Zhang *et al*. [29]	467.0	498.5
MCNN [30]	377.6	509.1
Cascaded-MTL [1]	322.8	341.4
Switching-CNN [23]	318.1	439.2
CP-CNN [26]	295.8	320.9
PACNN [17]	304.9	411.7
SANet [2]	258.4	334.9
CSRNet [11]	266.1	397.5
CSRNet*	264.0	398.1
Ours A	259.4	**317.6**
Ours B	**244.6**	361.2

Table 4: Comparisons on UCF_CC_50 [7] dataset.

Crowd Surveillance is our newly proposed dataset with 10,880 and 3,065 images for training / testing, as illustrated in Sec. 4. Similar to WorldExpo'10, we also provide ROI annotations that are used in the preprocessing. Table 5 demonstrates the comparisons of our method against MCNN [30], Switching-CNN [23] and CSRNet [11], and our method achieves the best results. Moreover, when we directly adopt the perspective map estimated by the PENet pre-trained on ShanghaiTech A (*Ours A*), we achieve 2.1 MAE gain over the baseline; and when we train the model end-to-end (*Ours B*), another 0.5 MAE gain over *Ours A* is achieved. We also provide visualization results of a test example in Fig. 6. It is seen that even perspective annotation is completely absent in Crowd Surveillance, the PENet is still able to provide reasonable perspective estimations, which supports its generalization ability. Such observations further validates our end-to-end training strategy described in Sec. 3.2.

Method	MAE	MSE
MCNN [30]	23.8	49.9
Switching-CNN [23]	16.9	33.2
CSRNet*	9.8	21.6
Ours A	7.7	16.4
Ours B	**7.2**	**15.6**

Table 5: Comparisons on Crowd Surveillance.
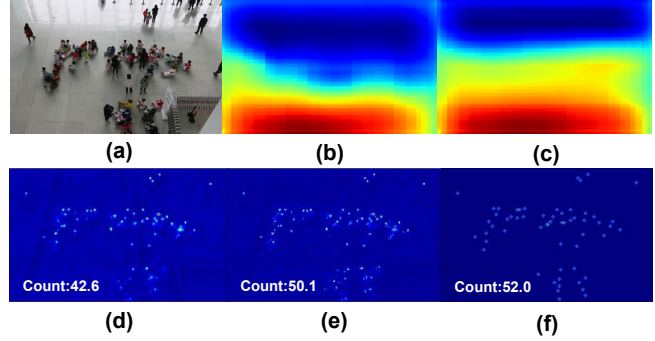

(a)　(b)　(c)
(d)　(e)　(f)

Figure 6: Visualization of a test example from Crowd Surveillance. (a) is the input; (b)(c) / (d)(f) are estimated perspective / density maps of Ours A and Ours B; and (f) is the ground truth.

5.3. Ablation Study

The evaluations against the state-of-the-arts above demonstrate the superiority of our PGC block and the end-to-end training strategy. We first conduct the experiment on choosing the appropriate value of K. Then we show the influence of the number of PGC blocks stacked in our network. Moreover, we demonstrate the feasibility of PGC block being an insertable component for existing network to improve performance. Finally, we also verify the importance of the pre-training of PENet in our method.

5.3.1 Influence of K

Table 6 lists the results of our method with a single PGC block but different K, *i.e.* the Gaussian filter size of Sec. 3.1. When K is small (*e.g.* ≤ 5), the PGC module is computationally light (~6ms for a 576×720 image, similarly hereinafter) but has poor performance; when K grows too large (*e.g.* ≥ 9), the performance starts to degrade with heavier computational burden (~16ms); when $K = 7$, it performs optimally with affordable efficiency (~10ms). We therefore adopt $K = 7$ in our following experiments.

K	MAE	MSE	K	MAE	MSE
$K = 1$	67.3	100.0	$K = 7$	**65.8**	98.0
$K = 3$	66.2	**97.7**	$K = 9$	66.3	98.8
$K = 5$	66.4	98.3	$K = 11$	66.4	97.4

Table 6: Influence of K on a single PGC block on ShanghaiTech Part A.

5.3.2 Influence of the Number of PGC Blocks

Table 7 shows the performance of our network when stacking different numbers of PGC blocks, as Fig. 2 shows. The performance increases with the number of PGC blocks stacked until reaching peak values (57.0 and 8.8 MAE on
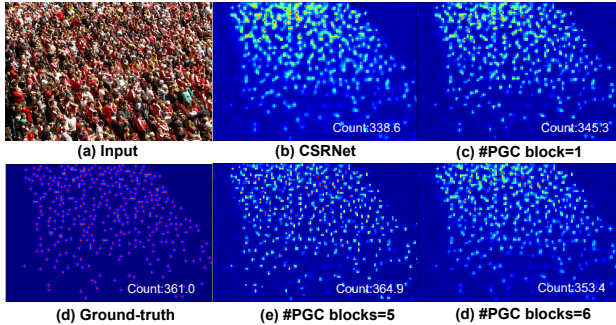
(a) Input	(b) CSRNet Count:338.6	(c) #PGC block=1 Count:345.3
(d) Ground-truth Count:361.0	(e) #PGC blocks=5 Count:364.9	(d) #PGC blocks=6 Count:353.4

Figure 7: Density maps predicted by stacking of different number of PGC blocks.

# of blocks	Part A		Part B	
	MAE	MSE	MAE	MSE
1	65.8	98.0	9.8	15.8
2	64.5	96.6	9.6	15.4
3	60.9	95.2	9.2	14.9
4	58.5	89.5	9.1	14.4
5	**57.0**	**86.0**	**8.8**	**13.7**
6	58.3	90.2	9.0	14.2

Table 7: Influence of the number of stacked PGC blocks in our method on ShanghaiTech.

Part A and Part B, respectively) with 5 blocks, and degrades afterwards. Fig. 7 demonstrates the predicted density maps when stacking serveral PGC blocks. It is seen that PGC gradually refines the density map till 5 PGC blocks. Too many PGC blocks(*e.g.*, 6) may lead to over-smoothing of features, resulting in worse estimation of density maps. We hence determine our final network architecture with 5 PGC blocks stacked, resulting in 100ms total time cost per 576×720 image in testing.

5.3.3 Extensibility to Another Backbone

In order to verify the extensibility of our PGC block to another backbone instead of VGG-16 [25] in CSRNet*, we adopt a truncated ResNet-101 [5] with the first 10 convolutional layers. It then appends PGC blocks (three in our experiment) and three extra convolutional layers are added afterwards to reduce the channel dimension to 1, making it compatible with the density map. Table 8 shows the comparison results on ShanghaiTech Part A / B, where *ResNet-101(backbone)* is the backbone itself, and *ResNet-101(PGC)* appends three PGC blocks. All models are trained for 500 epochs with the first 10 convolutional layers initialized by pre-trained weights on ImageNet [4]. For *ResNet-101(backbone)*, we get 109.6/26.2 MAE and 187.6/40.5 MSE. It is seen that when we stack three PGC blocks (*ResNet-101(PGC)*), we obtain the most performance gain of 19.9/7.6 MAE against *ResNet-101(backbone)*, reaching 89.7/18.6 MAE. Such observa-

tions validate the effectiveness and extensibility of the PGC block on another backbone.

Method	Part A		Part B	
	MAE	MSE	MAE	MSE
ResNet-101(backbone)	109.6	187.6	26.2	40.5
ResNet-101(PGC)	**89.7**	**148.4**	**18.6**	**30.9**

Table 8: Comparisons on the ResNet-101 backbone.

5.3.4 Importance of the pre-training of PENet

Although the end-to-end training strategy has been proposed, we still note that the PENet requires fair pre-training for accurate perspective estimation. To validate the necessity, we conduct two experiments without pre-training of PENet on UCF_CC_50 [7] and Crowd Surveillance, and compare the results in Table. 4 and Table. 5. As shown in Table 9, the results with PENet pre-training significantly outperform those without pre-training. Such performance margin is attributed to the confusing guidance of PENet on the spatially variant Gaussian smoothing of Sec. 3.1, since without pre-training, we observe that the output of PENet is still messy even after considerable epochs of training.

Method	UCF_CC_50		Crowd Surveillance	
	MAE	MSE	MAE	MSE
PENet(w/ pre-training)	**244.6**	**361.2**	**7.2**	**15.6**
PENet(w/o pre-training)	278.6	403.5	10.3	24.7

Table 9: The performances with / without pre-training of PENet on UCF_CC_50 [7] and Crowd Surveillance.

6. Conclusion

In this paper, we present a perspective-guided convolution network (PGCNet) for crowd counting. The key idea of PGCNet is the perspective-guided convolution, which functions as an insertable module that successfully handles the continuous intra-scene scale variation issue. We also propose a perspective estimation branch as well as its learning strategy, which is incorporated into our method to form an end-to-end trainable network, even without perspective map annotations. A new large scale dataset Crowd Surveillance is introduced as well to promote the researches in crowd counting. Experiments on four benchmark datasets show the superiority of our PGCNet against the state-of-the-arts.

Acknowledgement

References

[1] 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.

[2] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

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

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han. Body structure aware deep crowd counting. *IEEE Transactions on Image Processing*, 27(3):1049–1059, March 2018.

[7] 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.

[8] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. *arXiv preprint arXiv:1808.01050*, 2018.

[9] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010.

[10] 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.

[11] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, 2018.

[12] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2018.

[13] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. *arXiv preprint arXiv:1807.00601*, 2018.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[15] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. *arXiv preprint arXiv:1803.03095*, 2018.

[16] Miaojing Shi* Lu Zhang* and Qiaobo Chen. Crowd counting via sacle-adaptive convolutional neural network.

[17] Chao Xu Miaojing Shi, Zhaohui Yang and Qijun Chen. Perspective-aware cnn for crowd counting.

[18] Mohammadreza Mostajabi, Michael Maire, and Gregory Shakhnarovich. Regularizing deep networks by modeling and predicting label structure. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] 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.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[21] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 270–285, 2018.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] 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.

[24] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5245–5254, 2018.

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

[26] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1879–1888. IEEE, 2017.

[27] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302. ACM, 2015.

[28] 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.

[29] 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.

[30] 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.