Scale Aggregation Network for Accurate and Efficient Crowd Counting

Xinkun Cao[1], Zhipeng Wang[1], Yanyun Zhao[1,2], and Fei Su[1]

[1] School of Information and Communication Engineering
[2] Beijing Key Laboratory of Network System and Network Culture
Beijing University of Posts and Telecommunications, Beijing, China
{cc, wzpycg, zyy, sufei}@bupt.edu.cn

Abstract. In this paper, we propose a novel encoder-decoder network, called *Scale Aggregation Network (SANet)*, for accurate and efficient crowd counting. The encoder extracts multi-scale features with scale aggregation modules and the decoder generates high-resolution density maps by using a set of transposed convolutions. Moreover, we find that most existing works use only Euclidean loss which assumes independence among each pixel but ignores the local correlation in density maps. Therefore, we propose a novel training loss, combining of Euclidean loss and local pattern consistency loss, which improves the performance of the model in our experiments. In addition, we use normalization layers to ease the training process and apply a patch-based test scheme to reduce the impact of statistic shift problem. To demonstrate the effectiveness of the proposed method, we conduct extensive experiments on four major crowd counting datasets and our method achieves superior performance to state-of-the-art methods while with much less parameters.

Keywords: Crowd Counting · Crowd Density Estimation · Scale Aggregation Network · Local Pattern Consistency

1 Introduction

With the rapid growth of the urban population, crowd scene analysis [1,2] has gained considerable attention in recent years. In this paper, we focus on the crowd density estimation which could be used in crowd control for public safety in many scenarios, such as political rallies and sporting events. However, precisely estimating crowd density is extremely difficult, due to heavy occlusions, background clutters, large scale and perspective variations in crowd images.

Recently, CNN-based methods have been attempted to address the crowd density estimation problem. Some works [3,4,5,6] have achieved significant improvement by addressing the scale variation issue with multi-scale architecture. They use CNNs with different field sizes to extract features which are adaptive to the large variation in people size. The success of these works suggests that the multi-scale representation is of great value for crowd counting task. Besides, the crowd density estimation based approaches aim to incorporate the spatial

information of crowd images. As the high-resolution density maps contain finer details, we hold the view that it is helpful for crowd density estimation to generate the high-resolution and high-quality of density maps.

However, there exists two main drawbacks in recent CNN-based works. On the one hand, crowd density estimation benefits from the multi-scale representation of multi-column architecture, which uses multiple sub-networks to extract features at different scales. But the scale diversity is completely restricted by the number of columns (e.g. only three branches in multi-column CNN in [3]). On the other hand, only pixel-wise Euclidean loss is used in most works, which assumes each pixel is independent and is known to result in blurry images on image generation problems [7]. In [6], adversarial loss [8] has been applied to improve the quality of density maps and achieved good performance. Nevertheless, density maps may contain little high-level semantic information and the additional discriminator sub-network increases the computation cost.

To address these issues, we follow the two points discussed above and propose a novel encoder-decoder network, named *Scale Aggregation Network (SANet)*. The architecture of SANet is shown in Fig. 1. Motived by the achievement of Inception [9] structure in image recognition domain, we employ scale aggregation modules in encoder to improve the representation ability and scale diversity of features. The decoder is composed of a set of convolutions and transposed convolutions. It is used to generate high-resolution and high-quality density maps, of which the sizes are exactly same as input images. Inspired by [10], we use a combination of Euclidean loss and local pattern consistency loss to exploit the local correlation in density maps. The local pattern consistency loss is computed by SSIM [11] index to measure the structural similarity between the estimated density map and corresponding ground truth. The extra computation cost is negligible and the result shows it availably improves the performance.

We use Instance Normalization (IN) [12] layers to alleviate the vanishing gradient problem. Unfortunately, our patch-based model achieves inferior result when tested with images due to the difference between local (patch) and global (image) statistics. Thus, we apply a simple but effective patch-based training and testing scheme to diminish the impact of statistical shifts. Extensive experiments on four benchmarks show that the proposed method outperforms recent state-of-the-art methods.

To summarize, the main contributions of our work as follows:

– We propose a novel network, dubbed as *Scale Aggregation Network (SANet)* for accurate and efficient crowd counting, which improves the multi-scale representation and generates high-resolution density maps. The network can be trained end-to-end.
– We analyze the statistic shift problem caused by IN layers which are used to ease the training process. Then we propose a simple but effective patch-based train and test scheme to reduce its influence.
– We propose a novel training loss, combining Euclidean loss and local pattern consistency loss to utilize the local correlation in density maps. The former
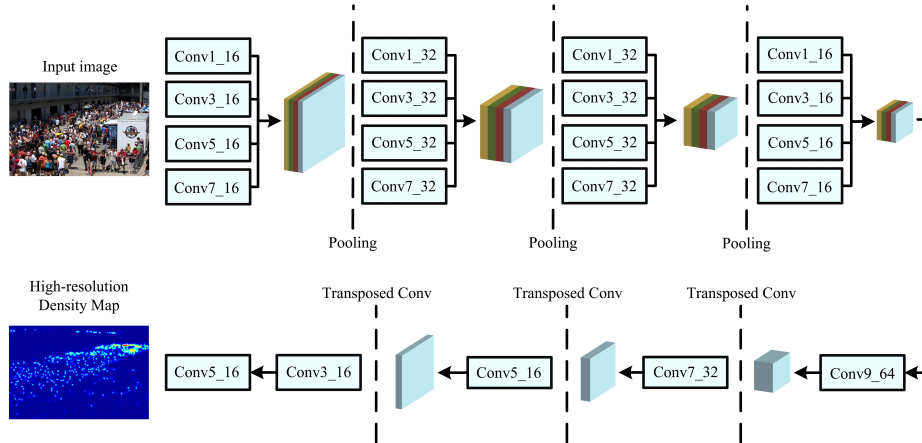
Fig. 1: The architecture of the SANet. A convolutional layer is denoted as "Conv(kernel size)_(number of channels)".

    loss limits the pixel-wise error and the latter one enforces the local structural similarity between predicted results and corresponding ground truths.
- Extensive experiments conducted on four challenging benchmarks demonstrate that our method achieves superior performance to state-of-the-art methods with much less parameters.

2 Related Works

A variety of methods have been proposed to deal with crowd counting task. They can be briefly summarized into traditional methods and CNN-based approaches.

2.1 Traditional Approaches

Most of the early works [13,14] estimate crowd count via pedestrian detection [15,16,17], which use body or part-based detector to locate people in the crowd image and sum them up. However, these detection-based approaches are limited by occlusions and background clutters in dense crowd scenes. Researchers attempted regression-based methods to directly learn a mapping from the feature of image patches to the count in the region [18,19,20]. With similar approaches, Idrees *et al.* [21] proposed a method which fuses features extracted with Fourier analysis, head detection and SIFT [22] interest points based counting in local patches. These regression-based methods predicted the global count but ignored the spatial information in the crowd images. Lempitsky *et al.* [23] proposed a method to learn a linear mapping between features and object density maps in local region. Pham *et al.* [24] observed the difficulty of learning a linear mapping and used random forest regression to learn a non-linear mapping between local patch features and density maps.

2.2   CNN-based Approaches

Due to the excellent representation learning ability of CNN, CNN-based works have shown remarkable progress for crowd counting. [25] introduced a comprehensive survey of CNN-based counting approaches. Wang *et al.* [26] modified AlexNet [27] for directly predicting the count. Zhang *et al.* [28] proposed a convolutional neural network alternatively trained by the crowd density and the crowd count. When deployed into a new scene, the network is fine-tuned using training samples similar to the target scene. In [29], Walach and Wolf made use of layered boosting and selective sampling methods to reduce the count estimation error. Different from the existing patch-based estimation methods, Shang *et al.* [30] used a network that simultaneously estimates local and global counts for whole input images. Boominathan *et al.* [31] combined shallow and deep networks for generating density map. Zhang *et al.* [3] designed multi-column CNN (MCNN) to tackle the large scale variation in crowd scenes. With similar idea, Onoro and Sastre [4] also proposed a scale-aware network, called Hydra, to extract features at different scales. Recently, inspired by MCNN [3], Sam *et al.* [5] presented Switch-CNN which trains a classifier to select the optimal regressor from multiple independent regressors for particular input patches. Sindagi *et al.* [32,6] explored methods to incorporate the contextual information by learning various density levels and generate high-resolution density maps. To improve the quality of density maps, they use adversarial loss to overcome the limitation of Euclidean loss. Li *et al.* [33] proposed CSRNet by combining VGG-16 [34] and dilated convolution layers to aggregate multi-scale contextual information.

However, by observing these recent state-of-the-art approaches, we found that: (1) Most works use multi-column architecture to extract features at different scales. As the issue discussed in Sec. 1, the multi-scale representation of this architecture might be insufficient to deal with the large size variance due to the limited scale diversity. (2) [5,32,6] require density level classifier to provide contextual information. However, these extra classifiers significantly increase the computations. In addition, the density level is related to specific dataset and is hard to be defined. (3) Most works use only pixel-wise Euclidean loss which assumes independence among each pixel. Though adversarial loss has shown improvement for density estimation, density maps may contain little high-level semantic information.

Based on the former observations, we propose an encoder-decoder network to improve the performance without extra classifier. Furthermore, we use a light-weight loss to enforce the local pattern consistency between the estimated density map and the corresponding ground truth.

3   Scale Aggregation Network

This section presents the details of the *Scale Aggregation Network (SANet)*. We first introduce our network architecture and then give descriptions of the proposed loss function.
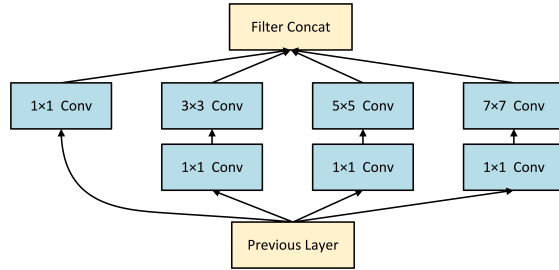
Fig. 2: The architecture of scale aggregation module.

3.1   Architecture

As shown in Fig. 1, we construct our SANet network based on two insights, i.e. multi-scale feature representations and high-resolution density maps. The SANet consists of two components: feature map encoder (FME) and density map estimator (DME). FME aggregates multi-scale features extracted from input image and DME estimates high-resolution density maps by fusing these features.

Feature Map Encoder (FME)   Most previous works use the multi-column architecture to deal with the large variation in object sizes due to perspective effect or across different resolutions. MCNN [3] contains three sub-networks to extract features at different scales. However, as the drawback mentioned in Sec. 1, the scale diversity of features is limited by the number of columns.

To address the problem, we propose an scale aggregation module to break the independence of columns with concatenation operation, as shown in Fig. 2. This module is flexible and can be extended to arbitrary branches. In this paper, we construct it by four branches with the filter sizes of 1×1, 3×3, 5×5, 7×7. The 1×1 branch is used to reserve the feature scale in previous layer to cover small targets, while others increase respective field sizes. The output channel number of each branch is set equal for simplicity. In addition, we add a 1×1 convolution before the $3 \times 3, 5 \times 5$ and 7×7 convolution layers to reduce the feature dimensions by half. These reduction layers are removed in the first scale aggregation module. ReLU is applied after every convolutional layer.

The FME of SANet is constructed by scale aggregation modules stacked upon each other as illustrated in Fig. 1, with 2×2 max-pooling layers after each module to halve the spatial resolution of feature maps. The architecture exponentially increases the possible combination forms of features and enhances the representation ability and scale diversity of the output feature maps. In this paper, we stack four scale aggregation modules. The stride of output feature map is 8 pixels $w.r.t$ the input image. Intuitively, FME might represent an ensemble of variable respective field sizes networks. The ensemble of different paths throughout the model would capture the multi-scale appearance of people in dense crowd, which would benefit the crowd density estimation.

Density Map Estimator (DME) While crowd density estimation based approaches take account of the spatial information, the output of most works is low-resolution and lose lots of details. To generate high-resolution density maps, we use a similar but deeper refinement structure to [6] as our DME, which is illustrated in Fig.1. The DME of our SANet consists of a set of convolutional and transposed convolutional layers. We use four convolutions to progressively refine the details of feature maps, with filter sizes from 9×9 to 3×3. And three transposed convolutional layers are used to recover the spatial resolution, each of which increases the size of feature maps by a factor 2. ReLU activations are added after each convolutional and transposed convolutional layers. Then, a 1×1 convolution layer is used to estimate the density value at each position. Since the values of density maps are always non-negative, we apply a ReLU activation behind the last convolution layer. Finally, DME generates the high-resolution density maps with the same size as input, which could provide finer spatial information to facilitate the feature learning during training the model.

Normalization Layers We observe a gradient vanishing problem which leads to non-convergence in training process when we combine FME and DME together. We attempt the Batch Normalzation [35] (BN) and Instance Normalization [12] (IN) to alleviate the problem, but get worse results when using BN due to the unstable statistic with small batchsize. Hence, we apply IN layers, which use statistics of each instance in current batch at training and testing, after each convolutional and transposed convolutional layers. However, our model trained by small patches gets inferior results when tested with whole images. We think it is caused by the statistical shifts. Considering the last 1×1 convolution layer and the preceding IN layer, for a d-dimensional vector $\mathbf{x} = (x_1...x_d)$ of input feature maps, the output is

$$y = ReLU \left(\sum_{i=0}^{d} w_i \cdot ReLU \left(\gamma_i \cdot \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_i \right) + b \right), \tag{1}$$

where w and b are weight and bias term of the convolution layer, γ and β are weight and bias term of the IN layer, μ and σ^2 are mean and variance of the input. The output is a weighted combination of the features which are normalized by the IN layer. Therefore, it is sensitive to the magnitude of features. But we find the difference of σ^2 are relatively large in some feature dimension when input patches or images. Then the deviation is amplified by square root and reciprocal function, and finally causes wrong density value. Since it is crucial to train the deep network with patches in consideration of speed and data augmentation, we apply a simple but effective patch-based training and testing scheme to reduce the impact of statistic shift problem.

3.2   Loss Function

Most existing methods use pixel-wise Euclidean loss to train their network, which is based on the pixel independence hypothesis and ignores the local correlation

of density maps. To overcome this issue, we use single-scale SSIM to measure the local pattern consistency and combine it with L_2 loss.

Euclidean Loss The Euclidean loss is used to measure estimation error at pixel level, which is defined as follows:

$$L_E = \frac{1}{N} \left\| F(X; \Theta) - Y \right\|_2^2 \tag{2}$$

where Θ denotes a set of the network parameters, N is the number of pixels in density maps, X is the input image and Y is the corresponding ground truth density map, $F(X; \Theta)$ denotes the estimated density map (we omit X and Θ for notational simplicity in later part). The Euclidean loss is computed at each pixel and summed over. Considering the size of input image may be different in a dataset, the loss value of each sample is normalized by the pixel number to keep training stable.

Local Pattern Consistency Loss Beyond the pixel-wise loss function, we also incorporate the local correlation in density maps to improve the quality of results. We utilize SSIM index to measure the local pattern consistency of estimated density maps and ground truths. SSIM index is usually used in image quality assessment. It computes similarity between two images from three local statistics, i.e. mean, variance and covariance. The range of SSIM value is from -1 to 1 and it is equal to 1 when the two image are identical. Following [11], we use an 11×11 normalized Gaussian kernel with standard deviation of 1.5 to estimate local statistics. The weight is defined by $W = \{W(\boldsymbol{p}) \mid \boldsymbol{p} \in \mathcal{P}, \mathcal{P} = \{(-5, -5), \cdots, (5, \ 5)\}\}$, where $\boldsymbol{p}$ is offset from the center and $\mathcal{P}$ contains all positions of the kernel. It is easily implemented with a convolutional layer by setting the weights to W and not updating it in back propagation. For each location $\boldsymbol{x}$ on the estimated density map F and the corresponding ground truth Y, the local statistics are computed by:

$$\mu_F(\boldsymbol{x}) = \sum_{\boldsymbol{p} \in \mathcal{P}} W(\boldsymbol{p}) \cdot F(\boldsymbol{x} + \boldsymbol{p}), \tag{3}$$

$$\sigma_F^2(\boldsymbol{x}) = \sum_{\boldsymbol{p} \in \mathcal{P}} W(\boldsymbol{p}) \cdot [F(\boldsymbol{x} + \boldsymbol{p}) - \mu_F(\boldsymbol{x})]^2, \tag{4}$$

$$\sigma_{FY}(\boldsymbol{x}) = \sum_{\boldsymbol{p} \in \mathcal{P}} W(\boldsymbol{p}) \cdot [F(\boldsymbol{x} + \boldsymbol{p}) - \mu_F(\boldsymbol{x})] \cdot [Y(\boldsymbol{x} + \boldsymbol{p}) - \mu_Y(\boldsymbol{x})], \tag{5}$$

where μ_F and σ_F^2 are the local mean and variance estimation of F, σ_{FY} is the local covariance estimation. μ_Y and σ_Y^2 are computed similarly to Equation 3, 4. Then, SSIM index is calculated point by point as following:

$$SSIM = \frac{(2\mu_F\mu_Y + C_1)(2\sigma_{FY} + C_2)}{(\mu_F^2 + \mu_Y^2 + C_1)(\sigma_F^2 + \sigma_Y^2 + C_2)}, \tag{6}$$

where C_1 and C_2 are small constants to avoid division by zero and set as [11]. The local pattern consistency loss is defined below:

$$L_C = 1 - \frac{1}{N} \sum_{\boldsymbol{x}} SSIM(\boldsymbol{x}), \tag{7}$$

where N is the number of pixels in density maps. L_C is the local pattern consistency loss that measures the local pattern discrepancy between the estimated result and ground truth.

Final Objective By weighting the above two loss functions, we define the final objective function as follows:

$$L = L_E + \alpha_C L_C, \tag{8}$$

where α_C is the weight to balance the pixel-wise and local-region losses. In our experiments, we empirically set α_C as 0.001.

4   Implementation Details

After alleviating the vanishing gradient problem with IN layers, our method can be trained end-to-end. In this section, we describe our patch-based training and testing scheme which is used to reduce the impact of statistic shift problem.

4.1   Training Details

In training stage, patches with 1/4 size of original image are cropped at random locations, then they are randomly horizontal flipped for data augmentation. Annotations for crowd image are points at the center of pedestrian head. It is required to convert these points to density map. If there is a point at pixel $\boldsymbol{x}_i$, it can be represented with a delta function $\delta(\boldsymbol{x} - \boldsymbol{x}_i)$. The ground truth density map Y is generated by convolving each delta function with a normalized Gaussian kernel G_σ:

$$Y = \sum_{\boldsymbol{x}_i \in \mathcal{S}} \delta(\boldsymbol{x} - \boldsymbol{x}_i) * G_\sigma, \tag{9}$$

where $\mathcal{S}$ is the set of all annotated points. The integral of density map is equal to the crowd count in image. Instead of using the geometry-adaptive kernels [3], we fix the spread parameter σ of the Gaussian kernel to generate ground truth density maps.

We end-to-end train the SANet from scratch. The network parameters are randomly initialized by a Gaussian distributions with mean zero and standard deviation of 0.01. Adam optimizer [36] with a small learning rate of 1e-5 is used to train the model, because it shows faster convergence than standard stochastic gradient descent with momentum in our experiments. The implementation of our method is based on the Pytorch [37] framework.

4.2   Evaluation Details

Due to the statistic shift problem caused by the IN layers, the input need to be consistent during training and testing. For testing the model trained based on patches, we crop each test sample to patches 1/4 size of original image with 50% overlapping. For each overlapping pixels between patches, we only reserve the density value in the patch of which the center is the nearest to the pixel than others, because the center part of patches has enough contextual information to ensure accurate estimation.

For crowd counting, the count error is measured by two metrics, Mean Absolute Error (MAE) and Mean Squared Error (MSE), which are commonly used for quantitative comparison in previous works. They are defined as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |C_i - C_i^{GT}|, \qquad MSE = \sqrt{\frac{1}{N} \sum_{i}^{N} |C_i - C_i^{GT}|^2}, \qquad (10)$$

where N is the number of test samples, C_i and C_i^{GT} are the estimated and ground truth crowd count corresponding to the ith sample, which is given by the integration of density map. Roughly speaking, MAE indicates the accuracy of predicted result and MSE measures the robustness. Because MSE is sensitive to outliers and it would be large when the model poorly performs on some samples.

5   Experiments

In this section, we first introduce datasets and experiment details. Then an ablation study is reported to demonstrate the improvements of different modules in our method. Finally, we give the evaluation results and perform comparisons between the proposed method with recent state-of-the-art methods.

5.1   Datasets

We evaluate our SANet on four publicly available crowd counting datasets: ShanghaiTech [3], UCF_CC_50 [21], WorldExpo'10 [28] and UCSD [38].

ShanghaiTech. The ShanghaiTech dataset [3] contains 1198 images, with a total of 330,165 annotated people. This dataset is divided to two parts: Part A with 482 images and Part B with 716 images. Part A is randomly collected from the Internet and Part B contains images captured from streets views. We use the training and testing splits provided by the authors: 300 images for training and 182 images for testing in Part A; 400 images for training and 316 images for testing in Part B. Ground truth density maps of both subset are generated with fixed spread Gaussian kernel.

WorldExpo'10. The WorldExpo10 dataset [28] consists of total 3980 frames extracted from 1132 video sequences captured with 108 surveillance cameras. The density of this dataset is relatively sparser in comparison to ShanghaiTech dataset. The training set includes 3380 frames and testing set contains 600 frames

Table 1: Ablation experiment results on ShanghaiTech Part A. Models are trained with patches and using only Euclidean loss unless otherwise noted

(a) **Modules**: Comparison the estimation error of different network configurations. MCNN∗ refers to our reimplementation

Model	MAE	MSE
MCNN [3]	110.2	173.2
MCNN∗	109.4	161.6
FME	90.5	**129.2**
MCNN+DME	**83.3**	134.7

(b) **Instance Normalization layers**: Estimation error of the models trained with or without IN layers. '-' indicates that the model fails to converge

Model	IN	MAE	MSE
MCNN+DME	×	83.3	134.7
MCNN+DME	✓	77.6	111.5
SANet	×	-	-
SANet	✓	**71.0**	**107.5**

(c) **Loss function and Test Scheme**: Estimation error of SANet trained with different loss functions and tested with different samples. L_E refers to Euclidean loss and L_C refers to local pattern consistency loss

Loss function	Test sample	MAE	MSE
L_E	image	116.8	180.4
L_E	patch	71.0	107.5
L_E, L_C	image	88.1	134.3
L_E, L_C	patch	**67.0**	**104.5**

from five different scenes and 120 frames per scene. Regions of interest (ROI) are provided for all scenes. We use ROI to prune the feature maps of the last convolution layer. During testing, only the crowd estimation error in specified ROI is computed. This dataset also gives perspective maps. We evaluate our method by ground truth generated with and without perspective maps. We follow experiment setting of [6] to generate density maps with perspective maps.

UCF_CC_50. The UCF_CC_50 dataset [21] includes 50 annotated crowd images. There is a large variation in crowd counts which range from 94 to 4543. The limited number of images make it a challenging dataset for deep learning method. We follow the standard protocol and use 5-fold cross-validation to evaluate the performance of proposed method. Ground truth density maps are generated with fixed spread Gaussian kernel.

UCSD. The UCSD dataset [38] consists of 2000 frames with size of 158×238 collected from surveillance videos. This dataset has relatively low density with an average of around 25 people in a frame. The region of interest (ROI) is also provided to ignore irrelevant objects. We use ROI to process the annotations. MAE and MSE are evaluated only in the specified ROI during testing. Following the train-test split used by [38], frames 601 through 1400 are used as training set and the rest as testing set. We generate ground truth density maps with fixed spread Gaussian kernel.
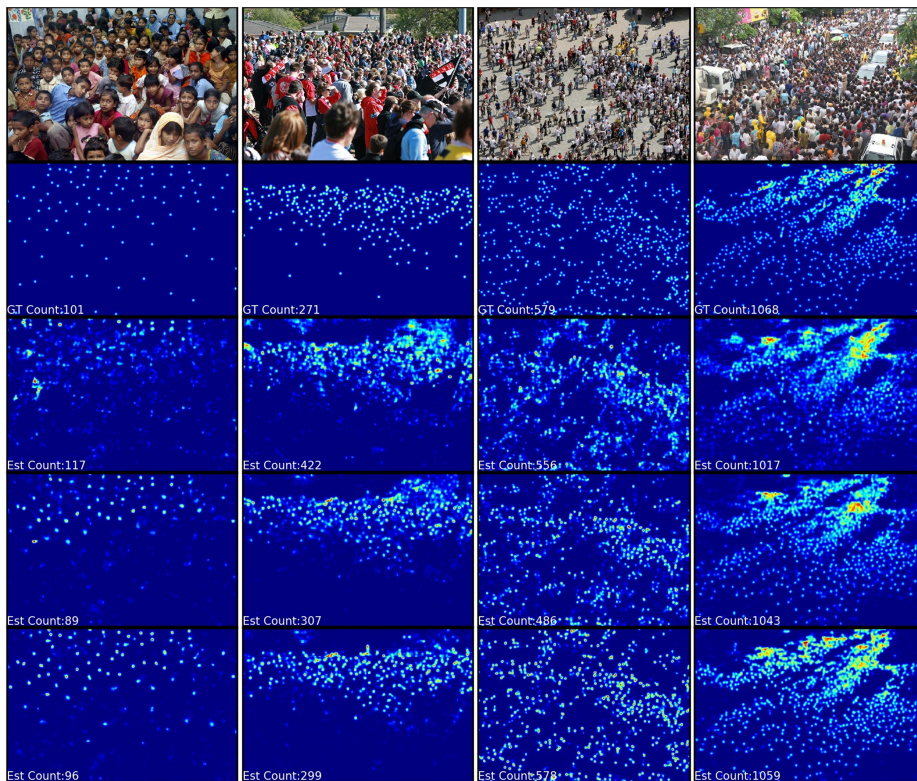
Fig. 3: Visualization of estimated density maps. First row: sample images from ShanghaiTech Part A. Second row: ground truth. Third row: estimated density maps by MCNN [3], which are resized to the same resolution as input images. Four row: estimated density maps by SANet trained with Euclidean loss only. Five row: estimated density maps by SANet trained with the combination of Euclidean loss and the local pattern consistency loss.

5.2   Ablation Experiments

We implement the MCNN and train it with the ground truth generated by fixed-spread Gaussian kernel. The result is slightly better than that reported in [3]. Based on the MCNN model, several ablation studies are conducted on ShanghaiTech Part A dataset. The evaluation results are reported in Table 1.

Architecture. We separately investigate the roles of FME and DME in SANet. We first append a 1×1 convolution layer to our FME to estimate density maps, which are 1/8 the size of input images. Both MCNN and FME output low-resolution density maps, but FME improves the scale diversity of features by the scale aggregation modules. Then, We combine DME with MCNN model to increase the resolution of density maps. With up-sampling by the transposed convolution layers in DME, the size of estimated density maps is the same as

input images. Compared with MCNN baseline, both FME and DME substantially reduce the estimation error. Table 1a shows that FME lowers MAE by 19.9 points and MSE by 32.4 points, while DME decreases 26.1 points of MAE and 26.9 points of MSE than baseline. This result demonstrate that the multi-scale feature representation and the high-resolution density map are extremely beneficial for the crowd density estimation task.

Instance normalization. Considering the vanishing gradient problem, we apply IN layers to both MCNN+DME and SANet. As illustrated in Table 1b, the IN layer can ease training process and boost the performance by a large margin. For MCNN+DME, IN layers decrease MAE by 5.7 points and MSE by 23.2 points. This result indicates that the model tends to fall into local minima without the normalization layers. Meanwhile, SANet with IN layers converges during training and achieves competitive results, with MAE of 71.0 and MSE of 107.5. The result would encourage attempts to use deeper network in density estimation problem.

Test scheme. We evaluated the SANet trained with patches by different input samples, i.e. images and patches. As shown in Table 1c, we can see that the SANet obtains promising results when testing with patches, but the performance is significantly dropped when testing with images. It verifies the statistic shift problem caused by IN layers. Therefore, it is indispensable to apply the patch-based test scheme.

Local pattern consistency loss. The result by using the combination of Euclidean loss and local pattern consistency loss is given in the Table 1c. We can observe that the model trained with the loss combination results in lower estimation error than using only L_E, which indicates this light-weight loss can improve the accuracy and robustness of model. Furthermore, the local pattern consistency loss significantly increases the performance when testing with images, which shows that the loss can enhance the insensitivity to statistic shift. We think it could smooth changes in the local region and reduce the statistical discrepancy between patches and images

Qualitative analysis. Estimated density maps from MCNN and our SANet with or without local pattern consistency loss on sample input images are illustrated in Fig. 3. We can see that our method obtains lower count error and generates higher quality density maps with less noise than MCNN. Moreover, the use of additional local pattern consistency loss further reduce the estimation error and improve the quality.

5.3   Comparisons with State-of-the-art

We demonstrate the efficiency of our proposal method on four challenging crowd counting datasets. Table 2, 3, 4, 5 report the results on ShanghaiTech, World-Expo'10, UCF_CC_50 and UCSD respectively. They show that the proposed method outperforms all other state-of-the-art methods from all tables, which indicates our method works not only in dense crowd images but also relatively sparse scene.

Table 2: Comparison with state-of-the-art methods on ShanghaiTech dataset [3]

Method	Part A		Part B	
	MAE	MSE	MAE	MSE
Zhang *et al.* [28]	181.8	277.7	32.0	49.8
MCNN [3]	110.2	173.2	26.4	41.3
Cascaded-MTL [32]	101.3	152.4	20.0	31.1
Huang *et al.* [39]	-	-	20.2	35.6
Switch-CNN [5]	90.4	135.0	21.6	33.4
CP-CNN [6]	73.6	106.4	20.1	30.1
CSRNet [33]	68.2	115.0	10.6	16.0
SANet(ours)	**67.0**	**104.5**	**8.4**	**13.6**

Table 3: Comparison with state-of-the-art methods on WorldExpo'10 dataset [28]. Only MAE is computed for each scene and then averaged to evaluate the overall performance

Method	Scene1	Scene2	Scene3	Scene4	Scene5	Avgerage
Zhang *et al.* [28]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [3]	3.4	20.6	12.9	13.0	8.1	11.6
Huang *et al.* [39]	4.1	21.7	11.9	11.0	3.5	10.5
Switch-CNN [5]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [6]	2.9	14.7	10.5	**10.4**	5.8	8.9
CSRNet [33]	2.9	**11.5**	**8.6**	16.6	3.4	8.6
SANet(ours) with perspective	2.8	14.0	10.2	12.5	3.5	8.6
SANet(ours) w/o perspective	**2.6**	13.2	9.0	13.3	**3.0**	**8.2**

As shown in Table 2, our method obtains the lowest MAE on both subset of ShanghaiTech. On WorldExpo'10 dataset, our approaches with and without perspective maps, both are able to achieve superior result compared to the other methods in Table 3. In addition, the method without perspective maps gets better result than using it and acquires the best MAE in two scenes. In Table 4, our SANet also attains the lowest MAE and a comparable MSE comparing to other eight state-of-the-art methods, which states our SANet also has decent performance in the case of small dataset. Table 5 shows that our method outperforms other state-of-the-art methods even in sparse scene. These superior results demonstrate the effectiveness of our proposed method.

As shown in Table 6, the parameters number of our proposed SANet is the least except MCNN. Although CP-CNN and CSRNet have comparable result with our method, CP-CNN has almost 75× parameters and CSRNet has nearly 17× parameters than ours. Our method achieves superior results than other state-of-the-art methods while with much less parameters, which proves the efficiency of our proposed method.

Table 4: Comparison with state-of-the-art methods on UCF_CC_50 dataset [21]

Method	MAE	MSE
Idrees *et al.* [21]	419.5	541.6
Zhang *et al.* [28]	467.0	498.5
MCNN [3]	377.6	509.1
Huang *et al.* [39]	409.5	563.7
Hydra-2s [4]	333.7	425.3
Cascaded-MTL [32]	322.8	341.4
Switch-CNN [5]	318.1	439.2
CP-CNN [6]	295.8	**320.9**
CSRNet [33]	266.1	397.5
SANet(ours)	**258.4**	334.9

Table 5: Comparison with state-of-the-art methods on UCSD dataset [38]

Method	MAE	MSE
Zhang *et al.* [28]	1.60	3.31
MCNN [3]	1.07	1.35
Huang *et al.* [39]	**1.00**	1.40
CCNN [4]	1.51	-
Switch-CNN [5]	1.62	2.10
CSRNet [33]	1.16	1.47
SANet(ours)	1.02	**1.29**

Table 6: Number of parameters (in millions)

Method	MCNN [3]	Switch-CNN [5]	CP-CNN [6]	CSRNet [33]	**SANet**
Parameters	0.13	15.11	68.4	16.26	**0.91**

6   Conclusion

In this work, we propose a novel encoder-decoder network for accurate and efficient crowd counting. To exploit the local correlation of density maps, we propose the local pattern consistency loss to enforce the local structural similarity between density maps. By alleviating the vanishing gradient problem and statistic shift problem, the model can be trained end-to-end. Extensive experiments show that our method achieves the superior performance on four major crowd counting benchmarks to state-of-the-art methods while with much less parameters.

Acknowledgement

References

1. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: Crowd analysis: a survey. Machine Vision and Applications **19**(5-6) (2008) 345–357
2. Li, T., Chang, H., Wang, M., Ni, B., Hong, R., Yan, S.: Crowded scene analysis: A survey. IEEE transactions on circuits and systems for video technology **25**(3) (2015) 367–386
3. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 589–597
4. Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision, Springer (2016) 615–629
5. Sam, D.B., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2017) 6
6. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid cnns. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE (2017) 1879–1888
7. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1125–1134
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
9. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions, IEEE (2015)
10. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for neural networks for image processing. IEEE Transactions on Computational Imaging (2017)
11. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4) (2004) 600–612
12. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. CoRR, abs/1703.06868 (2017)
13. Ge, W., Collins, R.T.: Marked point processes for crowd counting. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 2913–2920
14. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE (2008) 1–4
15. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE transactions on pattern analysis and machine intelligence **34**(4) (2012) 743–761
16. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32**(9) (2010) 1627–1645
17. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 1., IEEE (2005) 878–885

18. Chan, A.B., Vasconcelos, N.: Bayesian poisson regression for crowd counting. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 545–551
19. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: BMVC. Volume 1. (2012) 3
20. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: Digital Image Computing: Techniques and Applications, 2009. DICTA'09., IEEE (2009) 81–88
21. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 2547–2554
22. Ng, P.C., Henikoff, S.: Sift: Predicting amino acid changes that affect protein function. Nucleic acids research **31**(13) (2003) 3812–3814
23. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Advances in neural information processing systems. (2010) 1324–1332
24. Pham, V.Q., Kozakaya, T., Yamaguchi, O., Okada, R.: Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3253–3261
25. Sindagi, V.A., Patel, V.M.: A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognition Letters (2017)
26. Wang, C., Zhang, H., Yang, L., Liu, S., Cao, X.: Deep people counting in extremely dense crowds. In: Proceedings of the 23rd ACM international conference on Multimedia, ACM (2015) 1299–1302
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
28. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, IEEE (2015) 833–841
29. Walach, E., Wolf, L.: Learning to count with cnn boosting. In: European Conference on Computer Vision, Springer (2016) 660–676
30. Shang, C., Ai, H., Bai, B.: End-to-end crowd counting via joint learning local and global count. In: Image Processing (ICIP), 2016 IEEE International Conference on, IEEE (2016) 1215–1219
31. Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 2016 ACM on Multimedia Conference, ACM (2016) 640–644
32. Sindagi, V.A., Patel, V.M.: 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, IEEE (2017) 1–6
33. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. (2015) 448–456

36. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ICLR (2015)
37. Paszke, A., Chintala, S., Collobert, R., Kavukcuoglu, K., Farabet, C., Bengio, S., Melvin, I., Weston, J., Mariethoz, J.: Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, may 2017
38. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–7
39. Huang, S., Li, X., Zhang, Z., Wu, F., Gao, S., Ji, R., Han, J.: Body structure aware deep crowd counting. IEEE Transactions on Image Processing **27**(3) (2018) 1049–1059