Learning Spatial Awareness to Improve Crowd Counting

Zhi-Qi Cheng\textsuperscript{1,2,*}, Jun-Xiu Li\textsuperscript{1,3,*}, Qi Dai\textsuperscript{3}, Xiao Wu\textsuperscript{1}, Alexander G. Hauptmann\textsuperscript{2}
\textsuperscript{1}Southwest Jiaotong University, \textsuperscript{2}Carnegie Mellon University, \textsuperscript{3}Microsoft Research
\{zhiqic, alex\}@cs.cmu.edu, \{lijunxiu@my, wuxiaohk@home\}.swjtu.edu.cn, qid@microsoft.com

Abstract

The aim of crowd counting is to estimate the number of people in images by leveraging the annotation of center positions for pedestrians’ heads. Promising progresses have been made with the prevalence of deep Convolutional Neural Networks. Existing methods widely employ the Euclidean distance (i.e., L\textsubscript{2} loss) to optimize the model, which, however, has two main drawbacks: (1) the loss has difficulty in learning the spatial awareness (i.e., the position of head) since it struggles to retain the high-frequency variation in the density map, and (2) the loss is highly sensitive to various noises in crowd counting, such as the zero-mean noise, head size changes, and occlusions. Although the Maximum Excess over SubArrays (MESA) loss has been previously proposed by [16] to address the above issues by finding the rectangular subregion whose predicted density map has the maximum difference from the ground truth, it cannot be solved by gradient descent, thus can hardly be integrated into the deep learning framework. In this paper, we present a novel architecture called SPatial Awareness Network (SPANet) to incorporate spatial context for crowd counting. The Maximum Excess over Pixels (MEP) loss is proposed to achieve this by finding the pixel-level subregion with high discrepancy to the ground truth. To this end, we devise a weakly supervised learning scheme to generate such region with a multi-branch architecture. The proposed framework can be integrated into existing deep crowd counting methods and is end-to-end trainable. Extensive experiments on four challenging benchmarks show that our method can significantly improve the performance of baselines. More remarkably, our approach outperforms the state-of-the-art methods on all benchmark datasets.

1. Introduction

The problem of crowd counting is described in [16]. Different from visual object detection, it is impossible to provide bounding boxes for all pedestrians due to the extremely dense crowds. On the other side, when only the total crowd counts of the images are provided, the training process will become notably difficult since the spatial awareness is completely ignored. Therefore, to preserve as many spatial constraints as possible and reduce annotation cost, the previous work [16] started to only provide center points of heads and utilizes Gaussian distribution to generate ground truth density maps. It is worth noting that this annotation scheme is widely adopted by subsequent studies.

Existing crowd counting approaches mainly focus on improving the scale invariance of feature representation, including the multi-column networks [13, 38, 39, 42, 52, 56], scale aggregation modules [3, 47], and scale-invariant networks [9, 17, 20, 39, 45]. Despite the architectures of these methods are different, the L\textsubscript{2} loss function is employed by most of them. As a result, the spatial awareness in crowd image is largely ignored, though more scale information is embedded into their features.

We have examined three state-of-the-art approaches (i.e., MCNN [52], CSRNet [17], and SANet [3]) on four crowd counting datasets (i.e., ShanghaiTech [52], UCF_CC_50 [11], WorldExpo’10 [48], and UCSD [4]). Two examples are shown in Figure 1. Similar to [3, 19, 20], we observe that dense-crowd regions are usually underestimated, while sparse-crowd regions are overestimated. Such phenomenon is due to two main factors. First, the pixelwise L\textsubscript{2} loss struggles to retain the high-frequency variation

Figure 1: The L\textsubscript{2} loss function has difficulty in learning the spatial awareness and is sensitive to various noises in crowd counting, which will lead to a lower estimation in high-density regions (the first row of each example), and a higher estimation in low-density regions (the second row of each example). Note that the corresponding improvements of our method are shown in Figure 5.
in the density map: minimizing $L_2$ loss encourages finding pixel-wise averages of plausible solutions which are typically overly-smooth and thus have poor spatial awareness [15]. Second, $L_2$ loss is highly sensitive to typical noises in crowd counting, including the zero-mean noise, head size changes, and head occlusions. We take a simple statistics and show that the co-occurrence of zero-mean noise and overestimation could reach 96% (6,776 out of 7,044 testing images). We further find that almost all estimated density maps inaccurately predict the head positions or sizes when occlusion occurs, which could result in underestimation in high-density areas. Moreover, the generated ground truth density could also be imprecise due to the annotation error and the fixed variance in Gaussian kernel. It is noted that the corresponding improvements of our method are illustrated in Figure 5.

To fully utilize the spatial awareness, previous work [16] proposes a loss named Maximum Excess over SubArrays (MESA) to handle the above problems. Generally speaking, MESA loss attempts to find the rectangular subregion whose predicted density map has the maximum difference from the ground truth. It directly optimizes the counts of this subregion instead of the pixel-level density. Since the set of subregions could include the full image, MESA loss is an upper bound for the count estimation of the entire image. Besides, this loss is only sensitive to the spatial layout of pedestrians and is robust to various noises. However, the complexity of MESA loss function is extremely high. [16] utilizes Cutting-Plane optimization to obtain an approximate solution. Since this method cannot be solved by the conventional gradient descent, MESA loss has not been employed in any existing CNN-based approach.

Motivated by the MESA loss, in this paper we present a novel deep architecture called SPatial Awareness Network (SPANet) to retain the high-frequency spatial variations of density. Instead of finding the mismatched rectangular subregion as in MESA, the Maximum Excess over Pixels (MEP) loss is proposed to optimize the pixel-level subregion which has high discrepancy to the ground truth density map. To obtain such pixel-level subregion, the weakly-supervised ranking information [23] is exploited to generate a mask indicating the pixels with high discrepancies. We further devise a multi-branch architecture to leverage the full image for discrepancy detection by imitating the salience region detection [33, 50, 54], where patches with increasing areas are used for ranking. The proposed framework could be easily integrated into existing CNN-based methods and is end-to-end trainable.

The main contribution of this work is the proposed Spatial Awareness Network and Maximum Excess over Pixels loss for addressing the issue of crowd counting. The solution also provides the elegant views of what kind of spatial context should be exploited and how to effectively utilize such spatial awareness in crowd images, which are problems not yet fully understood in the literature.

2. Related Work

2.1. Detection-based Methods

The methods in this category use object detector to locate people in images. Given the individual localization of each person, crowd counting becomes trivial. There are two directions in this line, i.e., detection on 1) whole pedestrians [2, 7, 53] and 2) parts of pedestrians [8, 12, 18, 43]. Typically, local features [7, 18] are first extracted and then are exploited to train various detectors (e.g., SVM [18] and AdaBoost [41]). Though spatial information is well learned in these methods, they are not applicable in challenging situations, such as the high-density clogging crowds.

2.2. Regression-based Methods

Different from detection-based methods, regression-based approaches avoid the hard detection problem and estimate crowd counts from image features. Earlier methods [4, 5, 11, 28] usually predict the counts directly from the features, which will lead to poor performance as the spatial awareness is completely ignored. Later methods try to estimate the density map for counting [16, 26, 29], where the crowd count is obtained by integrating all pixel values over the density map. Though learning the density map somewhat provides the spatial information, their models still have difficulties in preserving the high-frequency variation in the density map.

2.3. CNN-based Methods

Deep CNN based crowd counting methods have shown very strong performance improvements over the shallow learning counterparts. Existing methods mainly focus on coping with the large variation in pedestrian scales, where many multi-column networks are extensively studied. A dual-column network is proposed by [1] to combine shallow and deep layers for estimating the count. Inspired by this work, a famous three-column network MCNN is proposed by [52], which employs different filters on separate columns to obtain features with various scales. Many works have improved MCNN [13, 38, 39, 42] to further enhance the scale adaptation. Sam et al. [32] introduce a switching structure, which uses a classifier to assign input image patches to appropriate columns. Recently, Liu et al. [19] propose a multi-column network to simultaneously estimate crowd density by detection and regression based models. Ranjan et al. [27] utilize a two-column network to iteratively train their model with images of different resolution.

There are a lot of other attempts to further improve the scale invariance, including 1) study on the fusion of various scale information [22, 40, 45, 46], 2) study on multi-blob based scale aggregation networks [3, 47], 3) design of
scale-invariant convolutional or pooling layers [9, 17, 20, 39, 45], and 4) study on the automated scale adaptive networks [30, 31, 49]. Typically, Li et al. [17] propose CSRNet that exploits dilated convolutional layers to enlarge receptive fields for boosting performance. Cao et al. [3] propose SANet to aggregate multi-scale features for more accurate crowd count. These two approaches have achieved state-of-the-art performance. Additionally, there also exist studies devoted to utilization of perspective maps [35], geometric constraints [21, 51], and region-of-interest (ROI) [20] to improve the counting accuracy.

The aforementioned methods utilize the Euclidean distance, i.e., $L_2$ loss to optimize the model. Although these methods can obtain scale-invariant features, their performances are still unsatisfactory since the spatial awareness is largely ignored. Note that, SANet [3] also tries to solve the problem of $L_2$ loss and adds local pattern consistency ($L_c$ loss) in the training phase. However, we find that $L_c$ still cannot learn the spatial context well. In our experiment, when integrating our MEP loss ($L_{mesa}$) into SANet, we achieve significant performance improvement. Our proposed MEP loss could fully utilize the spatial awareness, which is a key factor for the task of crowd counting.

3. Our Method

In this section, we first review the problem of crowd counting and two loss functions (i.e., center head positions annotated with a total of $c$ center points of pedestrians’ heads $P_i = \{P_1, P_2, \cdots, P_c\}$). Typically, the ground truth density map for each pixel $p$ in image $I_i$ is defined as $D_i$, $\forall p \in I_i, D_i(p) = \sum_{p \in P_i} N^g(p; \mu = P, \sigma^2)$, (1)

where $N^g$ is a Gaussian distribution. The number of people $c_i$ in image $I_i$ is equal to the sum of density values over all pixels as $\sum_{p \in I_i} D_i(p) = c_i$. With these training data, the aim of crowd counting task is to learn the predicted density map $D$ towards the ground truth density map $D^g$.

**MESA loss.** To make use of the spatial awareness in annotations (i.e., center head positions $P$), the previous work [16] has proposed the Maximum Excess over SubArrays (MESA) loss $L_{mesa}$ as follows,

$$L_{mesa}(D, D^g) = \frac{1}{N} \sum_{i=1}^{N} \max_{B \in \mathcal{B}} \left| \sum_{p \in B} D_i(p) - \sum_{p \in B} D^g(p) \right|,$$

(2)

where $\mathcal{B}$ is the set of all potential rectangular subregions in image. As illustrated in Figure 2, MESA loss tries to find the box subregion whose predicted density map has the maximum difference from the ground truth. It can be treated as an upper bound for the count estimation of the entire image, as $\mathcal{B}$ could include the full image. Besides, this loss is directly related to the counting objective instead of the pixel-level density, and is only sensitive to the spatial layout of pedestrians. In the 1D case, Kolmogorov-Smirnov distance [24] can be seen as a special case of $L_{mesa}$.

Despite the above merits, it is difficult to optimize the MESA loss due to the hard process of finding such subregion. One has to traverse all potential subregions to achieve this, which is obviously an impossible task in practical application. To solve it, previous approach [16] converts the optimization of MESA loss to a convex quadratic program problem with limited constraints and utilizes Cutting-Plane optimization to obtain an approximate solution. However, since this method cannot be solved by the traditional gradient descent, MESA loss has not been exploited in any existing CNN-based crowd counting methods.

**$L_2$ loss.** To facilitate the computation in deep frameworks, existing CNN-based methods [17, 27, 52] all directly use $L_2$ loss to minimize the difference between the estimated and ground truth density maps,

$$L_2(D, D^g) = \frac{1}{2N} \sum_{i=1}^{N} \sum_{p \in D} \|D_i(p) - D^g(p)\|^2_2. \quad (3)$$

Figure 2: Computation process of MESA loss. It is required to traverse all possible subregions and calculate the differences between their predicted density maps and the ground truth. Then the subregion with maximum difference is selected for optimization.

However, as discussed in Sec. 1, we reveal that $L_2$ loss can hardly retain the high-frequency variation in the density map, leading to the poor spatial awareness. Furthermore, it is also highly sensitive to typical noises in crowd counting, including the zero-mean noise, head size changes, and head occlusions. For example, existing methods always overestimate the density value in low-density areas and underestimate it in high-density regions.

3.2. Spatial Awareness Network

The proposed Spatial Awareness Network (SPANet) aims to leverage the spatial context for accurately predicting the density values. Instead of searching the mismatched
rectangular subregion as in MESA loss, which is the main obstacle for optimization, we try to find the pixel-level subregion \( S \) which has high discrepancy to the ground truth density map. Since there is not any annotation of such region, this problem is unsupervised and will still be significantly difficult to solve. Inspired by the recent weakly-supervised method \([23]\), we exploit an obvious ranking relation to achieve this, i.e., one patch of a crowded scene image is guaranteed to contain the same number or fewer persons than the original image. By sampling a pair of patches (where one is the sub-patch of the other), the network is optimized with the ranking objective by sampling two patches and outputs a new density map \( D_{k}^{pr} \). Then the two density maps are utilized to produce the subregion \( S_k \), which has high discrepancy to the ground truth. The density values within the generated \( S_k \) are erased in next branch to facilitate the latter optimization. In the end, \( K \) subregions from \( K \) branches are fused to form the final pixel-level subregion \( S \), which is exploited to calculate the Maximum Excess over Pixels (MEP) loss.

Figure 3 illustrates the framework of our proposed SPatial Awareness Network (SPANet). The input images are first fed into the backbone network to extract feature representations and output the estimated density maps \( D^{pr} \). A \( K \)-branch architecture is devised. In each branch \( k \), the network is optimized with the ranking objective by sampling two patches (one is sub-patch of the other) and outputs a new density map \( D_{k}^{pr} \). Then the two density maps are utilized to produce the subregion \( S_k \), which has high discrepancy to the ground truth. The density values within the generated \( S_k \) are erased in next branch to facilitate the latter optimization. In the end, \( K \) subregions from \( K \) branches are fused to form the final pixel-level subregion \( S \), which is exploited to calculate the Maximum Excess over Pixels (MEP) loss.

**Pixel-level Subregion Generation.** The subregion \( S \) indicates the area with high density discrepancy to the ground truth. Unfortunately, directly subtracting the predicted \( D^{pr} \) from the ground truth \( D^{gt} \) would make the problem go round in circles – the bias is usually large enough to prevent it from providing accurate region. Consequently, we turn to find the region with high changes along with the network training. It is natural that one can pick two density maps of the same image from different iterations. However, the obtained area only reflects the region that is already “revised”, which still seriously suffers from the poor spatial perception of the original \( L_2 \) loss. To this end, we exploit the weakly supervised ranking clues to produce the subregion. Instead of considering the pixel-level density, the ranking clue is directly related to the comparison of crowed counts.

In each branch \( k \), two parallel image patches are first sampled. As the feature maps of deep convolutional layers already contain rich location information, we treat the sampling process as the mask pooling operation on the density map. The strategy of selecting patches will be described later. Without loss of generality, suppose the two masks \( M^1_k \) and \( M^2_k \) are the 2-dimensional matrix with 0 or 1 (1 indicates the patch area), and \( M^1_{pr} \) is the sub-patch of \( M^1_k \). The crowd counts \( C(M^1_k) \) and \( C(M^2_k) \) under the masks \( M^1_{pr} \) and \( M^2_{pr} \) could be obtained by integrating the values of density map over individual mask, which could be implemented as the mask pooling as follows,

\[
C(M^1_k) = \sum_{p \in D^pr_k} (D^pr_k \odot M^1_k),
\]

\[
C(M^2_k) = \sum_{p \in D^pr_k} (D^pr_k \odot M^2_k),
\]

where \( \odot \) is the element-wise product, and \( p \) indicates the pixel on density map \( D^pr_k \). It is worth noting that we utilize the same predicted density map \( D^pr_k \) when calculating the counts for two masks, rather than generating individual maps at two consecutive iterations. The reason is that the density map \( D^pr_k \) is not restricted to be positive, thus pooling on the pair of patches could also provide the ranking information. We have conducted an experiment showing
that the two schemes have similar results. Besides, directly pooling on the same map is more efficient than the other.

With the assumption that $M_1^k$ is the sub-patch of $M_2^k$, the explicit constraint is that the number of people in $M_1^k$ is fewer than that in $M_2^k$. Therefore, we employ a pairwise ranking hinge loss $L_r$ to model such relationship, which is formulated as

$$L_r \left(C(M_1^k), C(M_2^k)\right) = \max \left(0, \ C(M_2^k) - C(M_1^k) + \xi \right),$$

where $\xi$ is a margin value that is set to the upper bound of the difference in the ground truth. The gradient of $L_r$ loss is calculated as

$$\nabla_{\theta} L_r = \begin{cases} 0, & \text{if } C(M_1^k) - C(M_2^k) + \xi \leq 0, \\ \nabla_{\theta} C(M_2^k) - \nabla_{\theta} C(M_1^k), & \text{otherwise}. \end{cases}$$

Once the network parameters $\theta$ are updated with $L_r$ by back-propagation, the renewed density map $\hat{D}^p_r$ estimated by the network is computed by

$$\hat{D}^p_r = \text{Conv} \left(I, \theta\right),$$

where $I$ is the input image, and $\text{Conv}(\cdot)$ refers to a forward pass of the network. Given the updated density map $\hat{D}^p_r$ and the old one $D^p_r$, the desired subregion $S_k$ is obtained by thresholding the difference $\nabla D^p_r$ between them, where $\nabla D^p_r = \vert \hat{D}^p_r - D^p_r \vert$. To make it differentiable, we utilize a Sigmoid thresholding function, and $S_k$ is given by

$$S_k = \frac{1}{1 + \exp \left(-\delta \left(\nabla D^p_r - \Sigma\right)\right)},$$

where $\Sigma$ is a threshold matrix with all elements being $\sigma$. $\delta$ is the parameter to ensure that the value of $S_k$ is approximately equal to 1 when $\nabla D^p_r(p) > \sigma$, otherwise 0.

**Multi-branch Architecture.** Note that in above section, only a pair of patches are sampled for generating the subregion. In principle, we hope that the full density map could be leveraged to provide more information. Instead of only sampling a small-large pair of patches, which may involve large bias error due to the large difference between two patches, we adopt a multi-branch architecture as shown in Figure 3. The bottom right corners of all patches are located at the same position, i.e., the bottom right corner of the density map. The area of patch is gradually enlarged along with the branches, until it reaches the size of full density map. Such design guarantees both the small bias error in each branch and the full utilization of training images.

To eliminate the influence of the detected subregion $S_k$ for better optimization in latter branches, we imitate the salience region detection [50] to erase the density values within $S_k$ in next branch, which is formulated as

$$D^p_{k+1} = D^p_{k+1} \odot (1 - S_k),$$

where 1 is the matrix with all elements being 1, and $\odot$ is the element-wise product.

### Maximum Excess over Pixels (MEP) loss

In the end, $K$ subregions ($S_1, S_2, ..., S_K$) are generated by the $K$ branches. The final desired pixel-level subregion $S$ is computed by simply combining them together as

$$S = \sum_{k=1}^{K} \{S_k\},$$

where $\sum$ indicates merging pixels with values close to 1 in all subregion masks $\{S_k\}$, rather than the direct summation. In practice, we take the maximum value at each pixel position from all masks. The final output $S$ is the mask that indicates the pixels which should be optimized. Based on that, our proposed MEP loss is then given by

$$L_{\text{mep}} \left(D_{\text{pr}}, D^{\text{gt}}\right) = \frac{1}{N} \sum_{i=1}^{N} \sum_{p \in S} D_{\text{pr},i}(p) - \sum_{p \in S} D^{\text{gt},i}(p).$$

### 3.3. Model Learning

Our SPANet could be easily integrated into existing crowd counting methods, which is equivalent to adding a pooling layer with different masks on the final convolutional layer. It is trained by sequentially optimizing the $K$ times ranking loss, MEP loss, and the original loss of existing methods. When calculating the original loss, the mask pooling layer is removed. The overall training objective is formulated as

$$L_{\text{global}} = \sum_{k=1}^{K} L_r + L_{\text{mep}} + L_{\text{vanilla}},$$

where $L_{\text{vanilla}}$ refers to the original loss of existing approach. In most cases, $L_{\text{vanilla}}$ is the $L_2$ loss. More details of the ground truth generation and data augmentation are described in supplementary material.

### 4. Experiment

#### 4.1. Experiment Settings

**Networks.** We evaluate our method by combining it with three networks, i.e., MCNN [52], CSRNet [17], and SANet [3]. The implementations of MCNN$^1$ and CSRNet$^2$ are from others, while SANet is implemented by us. In general, there are four main differences between them: (1) Different size of networks. Specifically, MCNN, SANet, and CSRNet are corresponding to small, medium, and large crowd counting networks. (2) Different architectures. MCNN and SANet are multi-column multi-blob networks, while CSRNet is a single column network. In addition, SANet uses the Instance Normalization (IN) layer and the deconvolutional layer, while CSRNet utilizes the dilated convolutional layer. (3) Different size of density maps. Density maps of MCNN and CSRNet are 1/4 and

$^1$https://github.com/svishwa/crowdcount-mcnn
$^2$https://github.com/leeyeehoo/CSRNet-pytorch/tree/master
Table 1: Performance comparison with the state-of-the-art methods on ShanghaiTech [52], UCF_CC_50 [11], and UCSD [48] datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Venue &amp; Year</th>
<th>ShanghaiTech A</th>
<th>ShanghaiTech B</th>
<th>UCF_CC_50</th>
<th>UCSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAE ↓</td>
<td>MSE ↓</td>
<td>MAE ↓</td>
<td>MSE ↓</td>
</tr>
<tr>
<td>Idrees et al. [11]</td>
<td>CVPR 2013</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>Zhang et al. [48]</td>
<td>CVPR 2015</td>
<td>181.8</td>
<td>277.7</td>
<td>32.0</td>
<td>49.8</td>
</tr>
<tr>
<td>CCNN [25]</td>
<td>ECCV 2016</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>Hydra-2s [25]</td>
<td>ECCV 2016</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>C-MTL [38]</td>
<td>AVSS 2017</td>
<td>101.3</td>
<td>152.4</td>
<td>20.0</td>
<td>31.1</td>
</tr>
<tr>
<td>SwitchCNN [32]</td>
<td>CVPR 2017</td>
<td>90.4</td>
<td>135.0</td>
<td>21.6</td>
<td>33.4</td>
</tr>
<tr>
<td>CP-CNN [39]</td>
<td>ICCV 2017</td>
<td>73.6</td>
<td>106.4</td>
<td>20.1</td>
<td>30.1</td>
</tr>
<tr>
<td>Huang et al. [10]</td>
<td>TIP 2018</td>
<td>- -</td>
<td>- -</td>
<td>20.2</td>
<td>35.6</td>
</tr>
<tr>
<td>SaCNN [49]</td>
<td>WACV 2018</td>
<td>86.8</td>
<td>139.2</td>
<td>16.2</td>
<td>25.8</td>
</tr>
<tr>
<td>ACSCP [34]</td>
<td>CVPR 2018</td>
<td>75.7</td>
<td>102.7</td>
<td>17.2</td>
<td>27.4</td>
</tr>
<tr>
<td>IG-CNN [31]</td>
<td>CVPR 2018</td>
<td>72.5</td>
<td>118.2</td>
<td>13.6</td>
<td>21.1</td>
</tr>
<tr>
<td>Deep-NCL [56]</td>
<td>CVPR 2018</td>
<td>73.5</td>
<td>112.3</td>
<td>18.7</td>
<td>26.0</td>
</tr>
<tr>
<td>MCNN [52]</td>
<td>CVPR 2016</td>
<td>110.2</td>
<td>173.2</td>
<td>26.4</td>
<td>41.3</td>
</tr>
<tr>
<td>CSRNet [17]</td>
<td>CVPR 2018</td>
<td>68.2</td>
<td>115.0</td>
<td>10.6</td>
<td>16.0</td>
</tr>
<tr>
<td>SANet [3]</td>
<td>ECCV 2018</td>
<td>67.0</td>
<td>104.5</td>
<td>8.4</td>
<td>13.6</td>
</tr>
<tr>
<td>MCNN+SPANet</td>
<td>- -</td>
<td>99.7</td>
<td>146.3</td>
<td>19.1</td>
<td>28.7</td>
</tr>
<tr>
<td>CSRNet+SPANet</td>
<td>- -</td>
<td>62.4</td>
<td>99.5</td>
<td>8.4</td>
<td>13.2</td>
</tr>
<tr>
<td>SANet+SPANet</td>
<td>- -</td>
<td>59.4</td>
<td>92.5</td>
<td>6.5</td>
<td>9.9</td>
</tr>
</tbody>
</table>

1/8 of original images, while SANet produces density maps with the same size as input images. (4) Different testing scheme. SANet is tested on image patches, while CSRNet and MCNN are tested on the whole images.

Learning settings. For MCNN and SANet, the parameters are randomly initialized by a Gaussian distribution with mean of 0 and standard deviation of 0.01. Adam optimizer [14] with a learning rate of 1e−5 is used to train the model. For CSRNet, the first ten convolutional layers are from pre-trained VGG-16 [37]. The other layers are initialized in the same way as MCNN. Stochastic gradient descent (SGD) with a fixed learning rate of 1e−6 is applied during the training.

Datasets. We evaluate our method on four datasets, including ShanghaiTech [52], UCF_CC_50 [11], WorldExpo’10 [48], and UCSD [4]. Typically, ShanghaiTech Part A is congested and noisy, while ShanghaiTech Part B is noisy but not highly congested. UCF_CC_50 consists of extremely congested scenes with heavy background noises. WorldExpo’10 and UCSD contain sparse crowd scenes. The scenes in WorldExpo’10 are noisier than UCSD.

Evaluation details. MCNN and CSRNet are tested on the whole images, while SANet is tested on image patches. Following previous works [17, 27, 52], Mean Absolute Error (MAE) and Mean Square Error (MSE) are used to evaluate the performance by

$$ MAE = \frac{1}{N} \sum_{i=1}^{N} |C_i - C_i^{gt}|, \quad MSE = \frac{1}{N} \sum_{i=1}^{N} (C_i - C_i^{gt})^2, $$

where $C_i$ is the estimated crowd count and $C_i^{gt}$ is the ground truth count of the $i$-th image. $N$ is the number of test images. Additionally, PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity) are utilized to measure the quality of density maps. For fair comparison, similar to [17], bilinear interpolation is employed to resize estimated density maps to the same size as input images.

4.2. Comparisons with State-of-the-art

Table 1 and 2 report the results of four challenging datasets. As a summary, our method significantly improves all baselines and outperforms the other state-of-the-art methods. This result fully demonstrates the effectiveness of our SP Anet, which could provide accurate density estimation on both dense and sparse crowd scenes, and can be applied to all CNN-based crowd counting networks.

On ShanghaiTech dataset, our SP Anet boosts MCNN, CSRNet, SANet with relative MAE improvements of 9.5%, 8.5%, 11.3% on Part A, and 27.7%, 20.8%, 22.7% on Part B. Noted that Part A is collected from the internet while Part B is from the busy streets and has more spatial constraints. Since our SP Anet can fully utilize spatial awareness, it brings more improvements on Part A, and 27.7%, 20.8%, 22.7% on Part B. Noted that Part A is collected from the internet while Part B is from the busy streets and has more spatial constraints. Since our SP Anet can fully utilize spatial awareness, it brings more improvements on Part B. On UCF_CC_50, SP Anet provides the relative MAE improvements of 22.5%, 7.6%, 10.0% for the three baselines. Noted that the improved MCNN is even comparable with other state-of-the-art methods. It clearly shows that SP Anet can handle the extremely dense-crowd scenes. Similar to the above two datasets, SP Anet also achieves significant improvements on UCSD and WorldExpo’10, verifying the effectiveness of our method on the sparse-crowd scenes.

4.3. Ablation Studies

Sampling positions. We first evaluate the impact of different starting positions when sampling patches for mask pooling. The results are listed in Table 3. We find that starting at the bottom is always better than the top, and the right is also better than the left. The possible reason is that it may be closely related to camera calibration. The results en-

3https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
4https://en.wikipedia.org/wiki/Structural_similarity
Table 2: Comparison with the state-of-the-art methods on WorldExpo'10 [4] dataset. Only MAE is computed for each scene and then averaged to evaluate the overall performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al. [46]</td>
<td>9.8</td>
<td>14.1</td>
<td>14.3</td>
<td>22.2</td>
<td>3.7</td>
<td>12.9</td>
</tr>
<tr>
<td>Huang et al. [10]</td>
<td>4.1</td>
<td>21.7</td>
<td>11.9</td>
<td>11.0</td>
<td>3.5</td>
<td>10.5</td>
</tr>
<tr>
<td>Switch-CNN [32]</td>
<td>4.6</td>
<td>15.7</td>
<td>10.0</td>
<td>11.0</td>
<td>5.9</td>
<td>9.4</td>
</tr>
<tr>
<td>SaCNN [49]</td>
<td>2.6</td>
<td>13.5</td>
<td>10.6</td>
<td>12.5</td>
<td>3.3</td>
<td>8.5</td>
</tr>
<tr>
<td>CP-CNN [39]</td>
<td>2.9</td>
<td>14.7</td>
<td>10.5</td>
<td>10.4</td>
<td>5.8</td>
<td>8.9</td>
</tr>
<tr>
<td>MCNN [52]</td>
<td>3.4</td>
<td>20.6</td>
<td>12.9</td>
<td>13.0</td>
<td>8.1</td>
<td>11.6</td>
</tr>
<tr>
<td>CSRNet [17]</td>
<td>2.9</td>
<td>11.5</td>
<td>8.6</td>
<td>16.6</td>
<td>3.4</td>
<td>8.6</td>
</tr>
<tr>
<td>SANet [3]</td>
<td>2.6</td>
<td>13.2</td>
<td>9.0</td>
<td>13.3</td>
<td>3.0</td>
<td>8.2</td>
</tr>
<tr>
<td>MCNN+SPANet</td>
<td>3.4</td>
<td>14.9</td>
<td>15.1</td>
<td>12.8</td>
<td>4.5</td>
<td>10.1</td>
</tr>
<tr>
<td>CSRNet+SPANet</td>
<td>2.6</td>
<td>11.1</td>
<td>8.9</td>
<td>13.5</td>
<td>3.3</td>
<td>7.9</td>
</tr>
<tr>
<td>SANet+SPANet</td>
<td>2.3</td>
<td>12.3</td>
<td>7.9</td>
<td>12.9</td>
<td>3.2</td>
<td>7.7</td>
</tr>
</tbody>
</table>

The Size of Estimated Density Maps

![The Size of Estimated Density Maps](image)

Figure 4: Ablation studies on ShanghaiTech Part A [52]. The left shows the branch number $K$ vs. MAE, and the right illustrates the size of estimated density maps vs. MAE, performed with MCNN.

Table 3: Ablation studies of patch sampling strategy, mask pooling strategy, and losses on ShanghaiTech Part A dataset [52].

<table>
<thead>
<tr>
<th>Configurations</th>
<th>MAE ↓</th>
<th>MSE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center point</td>
<td>101.2</td>
<td>153.3</td>
</tr>
<tr>
<td>Top left corner</td>
<td>101.5</td>
<td>153.7</td>
</tr>
<tr>
<td>Bottom left corner</td>
<td>100.7</td>
<td>149.2</td>
</tr>
<tr>
<td>Top right corner</td>
<td>100.5</td>
<td>149.4</td>
</tr>
<tr>
<td>Bottom right corner</td>
<td>99.7</td>
<td>146.3</td>
</tr>
<tr>
<td>Different density map</td>
<td>100.3</td>
<td>147.4</td>
</tr>
<tr>
<td>Same density map</td>
<td>99.7</td>
<td>146.3</td>
</tr>
<tr>
<td>$L_2$</td>
<td>110.2</td>
<td>173.2</td>
</tr>
<tr>
<td>$L_r + L_{r_{mep}}$</td>
<td>99.3</td>
<td>145.3</td>
</tr>
<tr>
<td>$L_2 + L_r$</td>
<td>107.2</td>
<td>164.5</td>
</tr>
<tr>
<td>$L_2 + L_{r_{mep}}$</td>
<td>99.7</td>
<td>146.3</td>
</tr>
<tr>
<td>Random</td>
<td>105.4</td>
<td>162.2</td>
</tr>
<tr>
<td>Grid Search</td>
<td>98.3</td>
<td>142.5</td>
</tr>
</tbody>
</table>

courage us to sample patches from the bottom right corner. Noted that the differences between these sampling schemes are quite small, which demonstrates the robustness of our method. Additionally, we also present the comparison of performing mask pooling on the same or different density maps in each branch, which is already discussed in Section 3.2 and Eq. (4). As shown in Table 3, the results of two strategies are similar. Due to the efficiency problem, we directly pool patches from the same density map.

Different losses/weights. We turn to evaluate the effect of different losses and weight schemes. As shown in Table 3, adding the ranking loss only provides slight improvement, while the significant improvement comes from the MEP loss. Besides, there is no significant difference whether $L_2$ is used. It demonstrates that our MEP loss can effectively learn spatial awareness to boost crowd counting. We further conduct experiments on two weight schemes: the random weight and the grid search with step 0.1. As shown in Table 3, our method is not sensitive to the weights. Even the grid search brings a very slight improvement.

Number of branches. We measure the performance of SPANet with different branch numbers $K$. As illustrated in Figure 4, the performance first improves but then drops with the increasing number of $K$. This observation is not surprising. On one side, small $K$ (e.g., $K = 1$) would involve large bias error due to the large difference between two patches. On the other side, large $K$ (e.g., $K = \frac{H}{16}$, where $H$ is the height of estimated density map) implies that the difference of two patches in each branch is very small, which cannot provide enough discrepancy for subregion generation. In experiments, $K$ is set to $\frac{H}{2}$ for MCNN/SANet and $\frac{H}{16}$ for CSRNet, which is determined via cross validation.

Size of estimated density maps. We further validate the effect of the size of estimated density maps. We add deconvolutional layers on top of the MCNN to increase the size of the estimated density maps. Eventually, two variants of MCNN are obtained, whose estimated density maps are of 1/2 and the same size as the input images, respectively. As shown in Figure 4, the performance is improved along with the size increase of density maps. The results indicate that predicting high-resolution density maps could bring considerable improvement.

4.4. Studies on Estimated Density Maps

We now evaluate the estimated density maps to verify whether our method can fully utilize spatial awareness. Table 4 summarizes the results. Our SPANet can significantly improve PSNR and SSIM across all baselines and datasets, which indicates that the quality of the generated density maps are significantly improved. To further verify that our method can indeed learn spatial awareness, we showcase the generated density maps of four examples from different methods in Figure 5. These four examples typically contain different crowd densities, occlusions, and scale changes. We can observe that the baseline models are always affected by the zero-mean noise, which leads to overestimation in low-density areas. In contrast, zero-mean noise is effectively suppressed in our SPANet. Besides, baseline models normally have an insufficient estimation for high-density areas, while ours can obtain a more accurate estimation for them. Noted that the ground truth itself is also generated with center points of pedestrians’ heads, which inherently contains inaccurate information. It means that our method is still unable to produce the same density map to the ground truth.

4.5. Studies on Learning Curves

Finally, we study the learning curves to further evaluate our method. Figure 6 shows the training and valida-
Figure 5: Comparisons of estimated density maps between baselines and our SPANet. ‘+’ indicates combining SPANet with baselines.

Figure 6: Learning Curves. Mean absolute error (MAE) on training and validation sets, vs. the number of training epochs of MCNN [52], CSRNet [17] and SANet [3] on ShanghaiTech Part A dataset [52].

Table 4: Density map quality comparison. Values on the left of ‘|’ are from original baselines, while values on the right of ‘|’ are results when integrating with the proposed SPANet.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MCNN</th>
<th>CSRNet</th>
<th>SANet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR ↑</td>
<td>SSIM ↑</td>
<td>PSNR ↑</td>
</tr>
<tr>
<td>ShanghaiTech-A</td>
<td>21.42</td>
<td>0.52</td>
<td>23.79</td>
</tr>
<tr>
<td>ShanghaiTech-B</td>
<td>23.43</td>
<td>0.78</td>
<td>27.02</td>
</tr>
<tr>
<td>UCF_CC_50</td>
<td>14.44</td>
<td>0.37</td>
<td>18.76</td>
</tr>
<tr>
<td>UCSD</td>
<td>17.43</td>
<td>0.75</td>
<td>20.02</td>
</tr>
<tr>
<td>WorldExpo’10</td>
<td>23.53</td>
<td>0.76</td>
<td>26.94</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper we present a novel deep architecture called SPatiatal Awareness Network (SPANet) for crowd counting, which is able to capture the spatial variations by finding the pixel-level subregion with high discrepancy to the ground truth. It could be integrated into all CNN-based methods and is end-to-end trainable. Experiments on four datasets and three various networks fully demonstrate that it can significantly improve all baselines and outperforms the state-of-the-art methods. It provides the elegant views of effectively using spatial awareness to improve crowd counting. In future work we will study how to preserve spatial awareness as much as possible in the ground truth generation.

Acknowledgements

This research was supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340, National Natural Science Foundation of China (61772436), Foundation for Department of Transportation of Henan Province, China (2019J-2-2), Sichuan Science and Technology Innovation Seedling Fund (2017RZ0015), China Scholarship Council (201707000083) and Cultivation Program for the Excellent Doctoral Dissertation of Southwest Jiaotong University (D-YB 201707).
References


[38] Vishwanath A. Sindagi and Vishal M. Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In Proceedings of International Conference on Advanced Video and Signal Based Surveillance, pages 1–6, 2017. 1, 2, 6


