Adaptive Density Map Generation for Crowd Counting

Jia Wan and Antoni Chan
Department of Computer Science, City University of Hong kong
jiawan1998@gmail.com, abchan@cityu.edu.hk

Abstract

Crowd counting is an important topic in computer vision due to its practical usage in surveillance systems. The typical design of crowd counting algorithms is divided into two steps. First, the ground-truth density maps of crowd images are generated from the ground-truth dot maps (density map generation), e.g., by convolving with a Gaussian kernel. Second, deep learning models are designed to predict a density map from an input image (density map estimation). Most research efforts have concentrated on the density map estimation problem, while the problem of density map generation has not been adequately explored. In particular, the density map could be considered as an intermediate representation used to train a crowd counting network. In the sense of end-to-end training, the hand-crafted methods used for generating the density maps may not be optimal for the particular network or dataset used. To address this issue, we first show the impact of different density maps and that better ground-truth density maps can be obtained by refining the existing ones using a learned refinement network, which is jointly trained with the counter. Then, we propose an adaptive density map generator, which takes the annotation dot map as input, and learns a density map representation for a counter. The counter and generator are trained jointly within an end-to-end framework. The experiment results on popular counting datasets confirm the effectiveness of the proposed learnable density map representations.

1. Introduction

Crowd counting is important task for understanding crowded scenes, and it can be used to prevent accidents caused by overcrowding and to estimate the crowd flows in stations. Given an image as input, the aim of crowd counting is to estimate the number of people in the image. Crowd counting is a difficult task since the scale of people varies dramatically in images, and the congested crowds frequently contains partial occlusions among people. One of the traditional methods is to detect each individual in the image, which does not work well in highly congested scenes

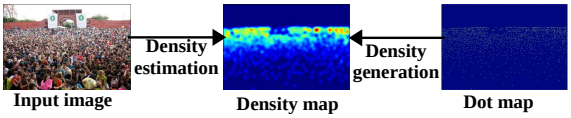

Figure 1. Counting by crowd density maps: the ground-truth crowd density map is generated from the ground-truth dot annotations of people. Given an input image, a model is trained to predict the density map, which is summed to obtain the predicted count. Current approaches treat the density map generation as a fixed intermediate representation, which is based on hand-crafted designs. In this paper, we propose to jointly learn the density estimator and the density generator.

[17, 39]. Another method is to directly estimate the final count based on hand-crafted features, which can only be applied to simple scenarios [4, 6].

Current state-of-the-art methods use crowd density maps to achieve superior counting performance [8, 7, 36]. Density maps are an *intermediate* representation, where the sum over any region in the density map indicates the number of people in the region. First, the density maps are generated from the dot annotations, where each dot indicates a person's location. Second, given the input image, algorithms are designed to predict the density map (see Figure 1), which is then summed to obtain the count. In this paper, we call these two steps density map *generation* and density map *estimation*, respectively. Most works focus on density map estimation and ignore density map generation. Many different deep networks have been proposed to improve density map estimation, e.g., using different kernel sizes [42] or image pyramids [14] to handle scale variations, or using context [33] or prior information [23] to handle occlusions. Although density map estimation is well-studied, the generation of density maps is often overlooked and uses handcrafted designs without adequate investigation and analysis. The simplest approach to obtain a density map is to convolve the annotation dot map with a Gaussian with fixed width [18], i.e., place a Gaussian on each dot. Other works scale the Gaussian bandwidth according to the scene perspective [41], or adaptively use the local congestion level (or distance to nearest neighbors) [42]. [41] uses human-shaped kernels, composed of two Gaussians, but is less popular since the body of the person is often occluded

in crowd images.

In practice, the method for generating the density maps is crucial for crowd counting. Improperly generated density maps may dramatically hurt the counting performance – the choice the kernel bandwidth or kernel shape used to generate the density map is often dataset dependent, and such choices often do not work across different datasets. In the era of deep learning, we may consider current density maps as a *hand-crafted intermediate representation*, which is used as a target for training deep networks to count. From the standpoint of end-to-end training, these hand-designed intermediate representations may not be optimal for the particular network architecture and particular dataset.

In this paper, we take the first step towards learnable density map representations, which are jointly trained with the density map estimator (counter). We propose two methods: 1) density map refinement, which is trained to improve existing traditional density maps; 2) adaptive density map generation, which learns a novel density map representation using the annotation dot map as input. Both methods are jointly trained with the density map estimator, yielding superior performance compared with using traditional density maps. The contributions of this paper are four-fold:

1. We study the impact of density maps on different datasets, and confirm through experiments that proper selection of density maps is essential for counting.
2. To improve manually-generated density maps, we propose to refine traditional density maps and achieve superior performance, which confirms that the quality of density maps can be improved.
3. We propose an adaptive density map generator, which takes the dot map as input, and produces a learnable density map representation. The density map generator and density map estimator (counter) are jointly trained.
4. With the proposed framework, we achieve state-of-the-art performance on ShanghaiTech A, ShanghaiTech B and UCF-QNRF without modifying the structure of the counter.

2. Related Work

Crowd counting algorithms can be divide into two categories: global regression and density estimation. Global regression directly estimates the final count from images, while density estimation first predicts a density map, which is then summed to obtain the final count. Since more spatial information is utilized in density estimation, the performance is usually better than global regression.

Most traditional counting algorithms are based on detection and global regression. [19] used head and shoulder detection for counting, but these detection-based algorithms will fail when people are highly occluded. Therefore, global regression algorithms were proposed to estimate crowd number without detection. Given an image as input, low-level features are extracted, from which a regression algorithm predicts the number of people [5, 4]. To improve the performance, multiple features are used in [12]. However, the performance of global counting is limited due to scale variation and occlusion in crowd images.

To better use spatial information of the people, [18] proposed crowd counting as a density map estimation problem, where the density map is an intermediate representation generated by the "dot" annotations of the people. To deal with scale variations, [42] proposed a multi-column convolutional network (MCNN) with different kernel sizes in each column. Instead of extracting multi-scale features, switch-CNN [29] chooses the column with the proper receptive field for the input image. Similarly, [1] proposed a tree-structured CNN to handle scale variations. SANet [3] is proposed to extract multi-scale features in all convolutional layers. Besides network structure, image pyramids are used in [14] to overcome scale variations.

Refinement-based approaches take an initial density map estimate and iteratively refine it to improve its accuracy. [26] propose a two-stage method, where the second stage estimates a high resolution density map from the low resolution density map predicted in the first stage, while [28] proposes a feedback mechanism to refine the prediction. Besides image-based refinement, a region-wise refinement algorithm is proposed in [22]. Related to refinement approaches are ensemble-based approaches, such as [35], which uses a CNN boosting algorithm, or [31], which uses multiple negative correlated regressors.

Finally, contextual information is also useful for crowd counting; [33] propose a contextual pyramid CNN (CP-CNN), while [40] uses temporal context. To exploit unlabelled data, [23] propose a ranking-based algorithm. To solve global counting, density estimation and localization simultaneously, the composition loss is proposed in [13]. Other works have also shown that crowd density maps are also useful for people detection and tracking in crowds [24, 15, 27]. A further survey of related work is in [34].

Although density map estimation has been researched for many years, density map generation has been largely overlooked. The current methods convolve the ground-truth dot annotation map with a Gaussian kernel with either fixed bandwidth [18], variable bandwidth based on the scene perspective [41], or adaptive bandwidth based on the crowdedness [42]. [13] composes multiple loss functions together, each using a ground-truth density map with a different fixed bandwidth. However, these bandwidth parameters are selected manually. In contrast to these hand-crafted methods, we propose a learnable density map generator, which is jointly trained with the counting algorithm.

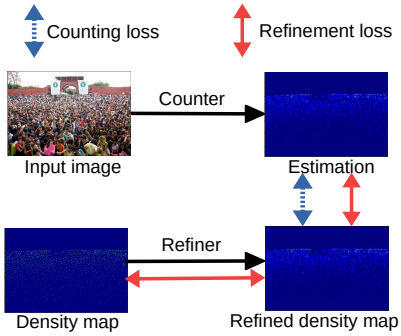

Figure 2. Density map refinement framework. The *Counter* is a network that estimates the density map of an input image. The *Refiner* is another network that takes a density map as input and produces a better density map as the ground truth to train the Counter. Both the Counter and Refiner are trained jointly.

3. Density Map Refinement

A traditional density map Y is generated by convolving the ground-truth dot map D, where each position with a person is marked as 1, with a Gaussian kernel,

$$Y = D * k_\sigma, \tag{1}$$

where k_σ is a 2D Gaussian kernel with bandwidth σ, and $*$ is 2D convolution. This is equivalent to placing a Gaussian on each dot annotation to obtain the density map

$$Y(p) = \sum_{\{p'|D(p')=1\}} \mathcal{N}(p|p', \sigma^2 I), \tag{2}$$

where p, p' are pixel locations in the image, and $\mathcal{N}(p|\mu, \Sigma)$ is a multivariate Gaussian with mean μ and covariance Σ. For adaptive kernels, the kernel bandwidth changes with location based on the crowdedness [42] or scene perspetive [41]. The counter is then trained using images and the corresponding density maps as the ground truth.

To confirm that traditional density maps can be improved to produce better counting performance, we first propose a density map refinement framework that jointly refines the density map and trains a counter from the refined density map (see Figure 2). Formally, let (X_i, Y_i) be the i-th image and traditional density map pair. We denote $f(X_i)$ as the predicted density map for image X_i, and $g(Y_i)$ as the refined density map for Y_i. The counter f and refiner g are jointly trained using a combined loss,

$$\mathcal{L} = \overbrace{\underbrace{\sum_{i=1}^{N} \|f(X_i) - g(Y_i)\|^2}_{\text{refinement loss}} + \alpha \|g(Y_i) - Y_i\|^2}^{\text{counting loss}}, \tag{3}$$

where N is the number of training pairs. The first term in (3) trains the counter to predict the refined density map, and vice versa, trains the refiner to produce density maps that favor the Counter's architecture. The second term in (3) constrains the refined density map to be close to the original density map Y_i, so that the global count and spatial distribution of the crowd is preserved. The joint training is summarized in Algorithm 1. Note that the refiner is only needed for training, i.e., to find the optimal intermediate representation – at inference time, only the counter is used to predict the density map from a novel image. Table 1 shows the architecture of the refiner. We use existing methods, such as MCNN [42], FCN-7c [14], SFCN [37] and CSRNet [20], for the counting network.

Algorithm 1 Training using density map refinement

Input: Set of image and density map pairs $\{(X_i, Y_i)\}_{i=1}^N$.
Initialize parameters of counter f and refiner g.
for $epoch = \{1, \ldots, N_e\}$ **do**
  for $i = \{1, \ldots, N\}$ **do**
    Estimate density map $f(X_i)$ by the counter.
    Produce refined ground truth $g(Y_i)$ by the refiner.
    Update counter f using the counting loss in (3).
    Update refiner g using the refinement loss in (3).
  end for
end for
Output: a counter and a refiner.

Table 1. The architectures of (top) the refiner and (bottom) the generator. C(K,S) is a convolution layer with K features and kernel size S. P is average pooling that decreases the spatial size by half. Each conv layer is followed by a ReLU, except the last layer.

Subnetwork	Architecture
Refiner	C(512,3)-C(512,3)-C(256,3)-C(128,3)-C(64,3)-C(1,3)
Self-attention	C(128,3)-C(32,3)-C(5,3)-Softmax
Fusion	C(128, 3)-P-C(32, 3)-P-C(8, 3)-P-C(1,3)-PReLU

4. Adaptive Density Map Generation

One disadvantage of the density map refiner proposed in Section 3 is it still depends on a hand-crafted density map as an input. Our experiments show that, although refinement can improve the accuracy, it still highly depends on the original density maps used. Therefore, in this section, we propose an adaptive density map generation framework, which *directly generates* the ground-truth density map from the ground-truth **dot** annotations. With this method, traditional density maps are not required and the whole system can be trained end-to-end without any intermediate steps.

The architecture of the proposed framework is shown in Fig. 3. Given a dot map as the input, the density map generator adaptively generate a density map based on the people distribution of the image by a self-attention fusion network. The learned density map is used to supervise the counter, and both the generator and counter are trained jointly.

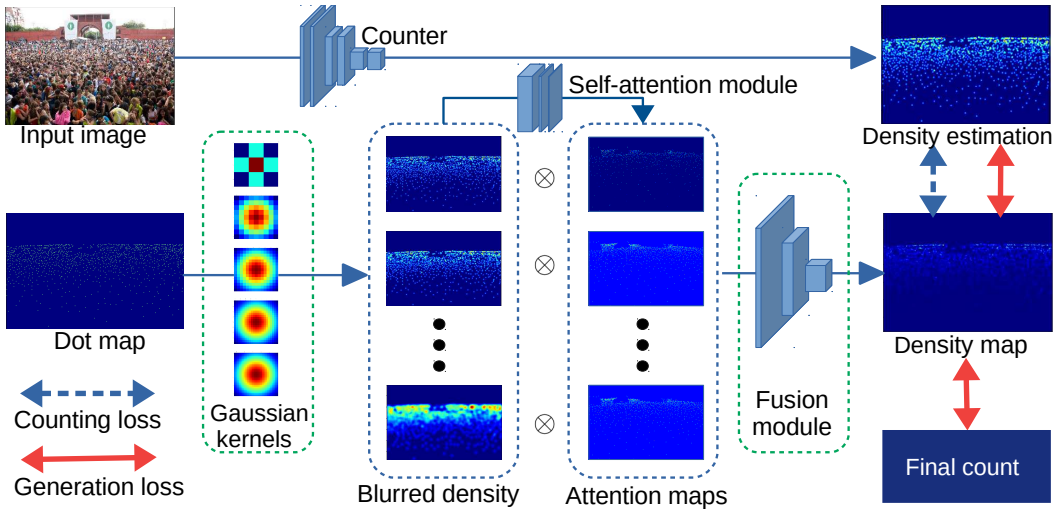

Figure 3. Density map generation framework. The input dot map is convolved with different Gaussian kernels, yielding a set of blurred density maps. The blurred density maps are adaptively masked using a self-attention module, and then passed through a fusion module to produce the final density map. The generated density map serves as the ground truth for training the density map estimator (counter).

4.1. Generation via Self-attention and Fusion

Given the dot map as input, the generation of the density map is divided into 3 steps: Gaussian blurring, self-attention, and fusion. First, the input dot map D_i is convolved with k Gaussian kernels with different bandwidths, resulting in a stack of k blurred density maps $B_i = \{B_i^j\}_j$,

$$B_i^j = D_i * k_{\sigma_j}, \tag{4}$$

which is equivalent to a convolutional layer with a different Gaussian kernel for each filter channel. Second, a self-attention module uses the blurred maps B_i as input to effectively select the best kernel size for each region, In particular, the attention map is

$$A_i = F_a(B_i), \tag{5}$$

where F_a is a small convolutional network (see Table 1), and each channel of A_i is an attention map for the corresponding blurred density map. Third, the blurred density maps are adaptively fused based on the attention maps,

$$M_i = F_f(A_i \otimes B_i), \tag{6}$$

where $\otimes$ is the pixel-wise multiplication, M_i is the final learned density map that is used to supervise the counter, and F_f is the fusion network (see Table 1).

4.2. Loss Function

Given a training set of images and corresponding ground-truth dot maps $\{(X_i, D_i)\}_{i=1}^N$, the density map generator and counter are jointly trained using the loss function,

$$\mathcal{L} = \underbrace{\overbrace{\sum_{i=1}^{N} \|\hat{M}_i - M_i\|^2}^{\text{counting loss}} + \beta(1^T M_i - 1^T D_i)^2}_{\text{refinement loss}} \tag{7}$$

where $\hat{M}_i$ is the density map predicted by the counter, M_i is the generated density map, and $1^T M$ is the sum over entries in M, i.e. the count from M. Similar to the refinement framework, the first term in (7) trains the counter to predict the generated density map, while also training the generator to produce density maps that the counter can predict well. The 2nd term in (7) encourages that the generated density maps have counts that are close to the ground-truth count.

In practice, we notice that the spatial information of the density map is well preserved when fixing the Gaussian kernels in the first stage of the generator (see experiments in Section 5.3.2). Thus we only use the global counting error to constrain the generated density maps. Algorithm 2 summarizes the training procedure for the counter and generator. The generator and dot maps are only used to train the counter. At test time, the counter predicts the density map from the input image.

5. Experiments

We present experiments using our proposed density map refinement and density map generator.

5.1. Experiment setup

We conduct experiments on four popular datasets, including ShanghaiTech (ShTech) A and B [42], WorldExpo [41], and UCF-QNRF [13]. ShanghaiTech A contains 482 crowd images with crowd numbers varying from 33 to 3139, and ShanghaiTech B contains 716 high-resolution images with crowd numbers from 9 to 578. WorldExpo evaluates the cross-scene crowd counting performance since the training images and testing images are from different scenes. UCF-QNRF is the most challenging dataset and contains 1535 high resolution images with very large

Algorithm 2 Training using density map generation

Input: Set of image and dot map pairs $\{(X_i, D_i)\}_{i=1}^{N}$.
Initialize parameters of counter and refiner.
for $epoch = \{1, \ldots, N_e\}$ **do**
  for $i = \{1, \ldots, N\}$ **do**
    Estimate density map $\hat{M}_i$ by the counter.
    Produce ground truth M_i by the generator.
    Update the counter using the counting loss in (7).
    {update generator every N_g epochs.}
    if $\mod (epoch, N_g) = 0$ **then**
      Update parameters of generator using (7).
    end if
  end for
end for
Output: a counter and a generator.

Table 2. Experimental results for density map refinement, evaluated using MAE. "w/o" means the baseline counter, and "w/ refine" means using the proposed density map refinement. The original density maps use "fixed" or "adaptive" Gaussian kernels.

		ShTech A		ShTech B	
Counter	Density Map	w/o	w/ refine	w/o	w/ refine
MCNN	Adaptive	103.3	**96.7**	**17.9**	18.0
	Fixed	**95.4**	102.3	17.3	17.3
FCN	Adaptive	95.4	**92.8**	**16.0**	16.7
	Fixed	90.7	**89.9**	18.8	**15.2**
SFCN	Adaptive	73.1	**70.5**	9.7	**9.0**
	Fixed	70.8	**67.8**	9.9	**9.3**
CSRNet	Adaptive	66.4	**64.2**	10.6	**9.2**
	Fixed	67.8	**66.9**	12.1	**11.1**

crowds. Methods are evaluated using mean absolute error (MAE) and root mean squared error (RMSE):

$$MAE = \frac{1}{N}\sum_i |\hat{y}_i - y_i|, \quad RMSE = \sqrt{\frac{1}{N}\sum_i \|\hat{y}_i - y_i\|^2},$$

where N is the number of samples and $\hat{y}_i$, y_i are the predicted and ground truth counts.

Our baseline counters include CSRNet [20], SFCN [37], MCNN [42], and FCN [14]. Their training procedures follow the original papers: SGD is used to train CSRNet with learning rate set to 5e-7; The Adam optimizer [16] is used to train SFCN with learning rate 1e-5; FCN and MCNN are trained with Adam with learning rate of 1e-5. For the refinement network, Adam optimizer is used for training and the learning rate is set to 1e-5. Different input density maps are generated following [20]. The fixed bandwidth is set to 16 and the hyper-parameter α is set to 1. For the generation network, Adam optimizer is used for training with learning rate of 1e-7 and $\beta = 1$.

5.2. The Importance of Proper Density Maps

We first show that different density maps may produce different performance when they serve as the ground truth for training a counter, and that these results can be improved using density map refinement. Two types of density maps are used: fixed kernel (bandwidth 16), and adaptive kernels [20]. The experiments are performed on ShTech A and B.

Table 3. Experimental results of density map generation (MAE). σ is the bandwidth for the fixed kernel.

Counter	Density Map	ShTech A	ShTech B
MCNN	Fixed kernel (σ=16)	95.4	18.7
	Fixed kernel (σ=4)	96.0	17.9
	Adaptive kernel	103.3	17.9
	Generator (ours)	**93.5**	**17.7**
FCN	Fixed kernel (σ=16)	90.7	18.8
	Fixed kernel (σ=4)	88.9	**13.8**
	Adaptive kernel	95.4	16.0
	Generator (ours)	**87.1**	13.9
SFCN	Fixed kernel (σ=16)	70.8	9.9
	Fixed kernel (σ=4)	70.8	10.6
	Adaptive kernel	73.1	9.7
	Generator (ours)	**68.4**	**8.4**
CSRNet	Fixed kernel (σ=16)	67.8	12.1
	Fixed kernel (σ=4)	70.1	9.5
	Adaptive kernel	66.4	10.6
	Generator (ours)	**64.7**	**8.1**

The results are presented in Table 2. The effectiveness of traditional density maps depends on the method and dataset (see "w/o" columns in Table 2). For example, on ShTech A, a fixed kernel is better for MCNN and FCN, but an adaptive kernel is better for CSRNet, while all most methods (except MCNN) do better with adaptive kernels on ShTech B. Looking at the results using refinement ("w/" columns), the counting performance of CSRNet and SFCN are always boosted when jointly training the density map refinement and counting tasks. However, the performance gain for MCNN is limited, compared with CSRNet, which suggests that the proposed density refinement framework requires a strong baseline counter. The strong counter's output is more accurate, and thus the refined density map does not need to be modified significantly, thus preserving the accuracy of the original density maps.

These results suggest that there is room to improve the manually-designed density maps. Unfortunately, the selection of traditional density maps depends on the dataset and counting network, which requires manual effort to tune. For this reason, we have proposed adaptive density map generation to learn to generate effective density maps directly from the annotation dot maps.

5.3. Density Map Generation

We next conduct experiments on the effectiveness of our proposed generation framework, which is jointly trained with a counting network. Table 3 compares the counting performance when using traditional density maps and our generated density maps. Almost all counters trained with the proposed generation framework achieve better perfor-

mance than both types of traditional density maps. The exception is FCN on ShTech B, where our generated density maps performs similarly to fixed kernels with bandwidth 4. Note that the generation framework works better on stronger baselines (CSRNet/SFCN) than weaker ones (MCNN/FCN). A possible reason is that a weak baseline introduces more noise when training the generator.

We visualize the generated density maps for two typical images in Figure 4. The traditional density maps using a fixed kernel is too smooth, while those produced by adaptive kernels are too sharp. The density maps from the proposed generation framework can adapt to the people distribution of the image, thus are more effective for the training of counter. We also visualize the difference between generated density maps trained with different counters in Figure 5. Specifically, we show the difference maps between density maps generated by CSRNet and the other networks. Since the receptive field of CSRNet is larger than FCN and MCNN, the generated density maps for CSRNet are more spread out (smoother) than those for FCN/MCNN, as shown in Figs. 5 (a) and (b). SFCN and CSRNet have similar receptive field size, so the smoothness of the generated density maps are also similar. Since SFCN utilizes a spatial CNN, its generated density maps are spatially moved from that of CSRNet, see Figure 5 (c). The visualization shows that the proposed framework can learn density maps that adapt to different architectures.

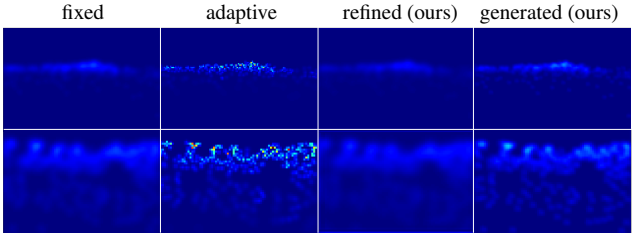

Figure 4. Density maps using traditional methods (fixed, adaptive) and learned methods (refined, generated).

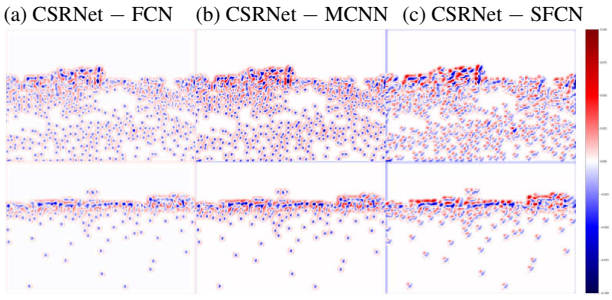

Figure 5. Comparison of generated density maps trained using CSRNet versus other counters (FCN, MCNN, SFCN). The difference map between the density map generated for CSRNet and for other methods is shown for 2 examples.

Table 4. Ablation study on ShanghaiTech A (MAE) for the density map generator using different loss functions and fixed/learned initial blurring kernels.

	global loss	local loss	hard norm
(a) fixed initial kernels	**64.7**	68.8	112.8
(b) learned initial kernels	74.7	73.0	101.6

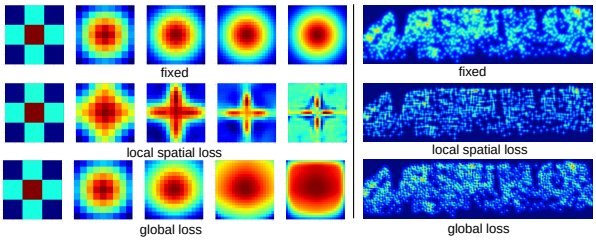

Figure 6. (left) fixed initial kernels, and trained kernels using local and global loss; (right) corresponding density maps.

5.3.1 Ablation Study: Loss Function

We run an ablation study on the choice of loss function for training the density map generator. We replace the global count loss in (7) with a local spatial loss, which is the average counting error over image patches. Instead of the counting loss, we also consider hard normalization of the generated density map so that it sums to the ground-truth count. The results of the ablation study are presented in Table 4 (a) – using the global loss term outperforms the spatial loss and hard normalization.

5.3.2 Ablation Study: Initial Blurring Kernels

During the training of generator, we fix the initial Gaussian kernels used to generate the set of blurred density maps. In this ablation study, we consider making these initial kernels learnable. Table 4 (b) presents the results when the initial kernels are learned using different loss functions. The counter/generator trained with fixed initial kernels have better counting performance than those with learnable initial kernels.

Figure 6 visualizes the fixed and learned initial kernels. The learnable kernels dramatically change compared to the initial Gaussian kernels. When using the spatial loss, the learned kernels become "plusses", which produce minimal leakage when an annotation is just outside the boundary of a spatial region. When using the global loss, the kernels expand to fill the whole convolution filter, which obfuscates spatial information in the generated density map.

5.3.3 Ablation Study: Self-attention Fusion

To confirm the effectiveness of the self-attention module (self-att), we compare it with three variations: 1) direct fusion without the attention module; 2) image-based attention (image-att), which uses the input image to generate attention; 3) naive fusion which sum the blurred maps directly.

Table 5. Ablation study of self-attention module on ShanghaiTech A. MAE↓ is used as the metric.

	self-att	image-att	direct-fusion	naive-fusion
MAE↓	**64.7**	66.9	67.5	68.6

As shown in Table 5, the self-attention module is more effective than these three variants. Direct-fusion ignores the spatial distribution of people, while image-att cannot handle the additional noise from the input image, such as distractor objects (trees) and backgrounds. We also visualize the attention maps and the blurred maps after attention in Figure 7. The density of the centre of a person comes from the small bandwidth map, while the boundary area comes from larger bandwidth maps. The visualization indicates that the small bandwidth maps concentrate on location of people while large bandwidth maps pay attention to the boundary of people.

5.3.4 Ablation Study: Generalization Ability

To evaluate the generalization ability of learned density maps, we conducted an experiment on ShanghaiTech A using the generated density maps trained for CSRNet as the ground-truth density maps to train MCNN, FCN, and SFCN. The results are shown in Table 6, where "Generator (jointly trained)" is the generator-estimator jointly trained together and "Generator for CSRNet" uses the density maps generated for CSRNet to train other estimators. The results show that density maps generated for one estimator (CSRNet) do not generalize well to other estimators (MCNN, FCN, SFN), as the error for "Generator for CSRNet" is higher than the jointly-trained generator, and in general similar to fixed/adaptive kernels. This demonstrates that the generated density maps are matching particular properties (e.g., receptive field size, network depth) of the estimators to improve the counting accuracy. In this case, as CSRNet is a large complex network, the density maps for CSRNet might be too complex for simpler networks (MCNN, FCN) to predict correctly.

Density map	MCNN	FCN	SFCN
Fixed kernel (σ=16)	95.4	90.7	70.8
Fixed kernel (σ=4)	96.0	88.9	70.8
Adaptive kernel	103.3	95.4	73.1
Generator (jointly trained)	93.5	87.1	68.4
Generator for CSRNet	97.6	89.0	73.2

Table 6. The experimental results (MAE) on ShanghaiTech A of the generalization ability of generated density maps.

5.4. Comparison with state-of-the-art

We compare our proposed density map refinement and generation frameworks with state-of-the-art methods on ShanghaiTech A, ShanghaiTech B, UCF-QNRF, and WorldExpo. Here we use CSRNet [20] as the baseline counting model. The experimental results are shown in Tables 7, 8, 9, and 10.

Table 7. Experimental results on ShanghaiTech A. MAE and RMSE are used to evaluate the performance.

Method	MAE ↓	MSE ↓
Cross-scene [41]	181.8	277.7
MCNN [42]	110.2	173.2
FCN [25]	126.5	173.5
Cascaded-MTL [32]	101.3	152.4
Switching-CNN [29]	90.4	135.0
CP-CNN [33]	73.6	106.4
ASACP [30]	75.7	102.7
Top-Down [28]	97.5	145.1
L2R [23]	73.6	112.0
IG-CNN [26]	72.5	118.2
ic-CNN [26]	68.9	117.3
SANet (patch) [3]	67.0	104.5
SANet (image) [3]	88.1	134.3
SCNet [38]	71.9	117.9
Spatial-Aware [22]	69.3	**96.4**
Image Pyramid [14]	80.6	126.7
CSRNet [20]	68.2	115.0
Ours (refinement)	**64.2**	99.7
Ours (generation)	64.7	97.1

Table 8. Experimental results on ShanghaiTech B. '†' means using Res101 as the backbone [37].

Method	MAE↓	MSE↓
Cross-scene [41]	32.0	49.8
MCNN [42]	26.4	41.3
FCN [25]	23.76	33.12
Cascaded-MTL [32]	20.0	31.1
Switching-CNN [29]	21.6	33.4
CP-CNN [33]	20.1	30.1
DecideNet [21]	20.75	29.42
ASACP [30]	17.2	27.4
Top-Down [28]	20.7	32.8
L2R [23]	14.4	23.8
IG-CNN [1]	13.6	21.1
ic-CNN [26]	10.7	16.0
SANet [3]	8.4	**13.6**
Spatial-Aware [22]	11.1	18.2
Image Pyramid [14]	10.2	18.3
SFCN†[37]	8.9	14.3
CSRNet [20]	10.6	16.0
Ours (refinement)	9.1	14.4
Ours (generation)	**8.1**	**13.6**

On the ShanghaiTech A dataset, both the proposed refinement and generation frameworks achieve superior performance compared with state-of-the-arts on MAE. However, the MSE is slightly behind Spatial-Aware [22], since our proposed method is a simple one stage estimation, while [22] iteratively refines the predicted density map. The proposed refinement/generation methods also outperforms the baseline model, CSRNet [20], which further confirms the effectiveness of learning the density map representation.

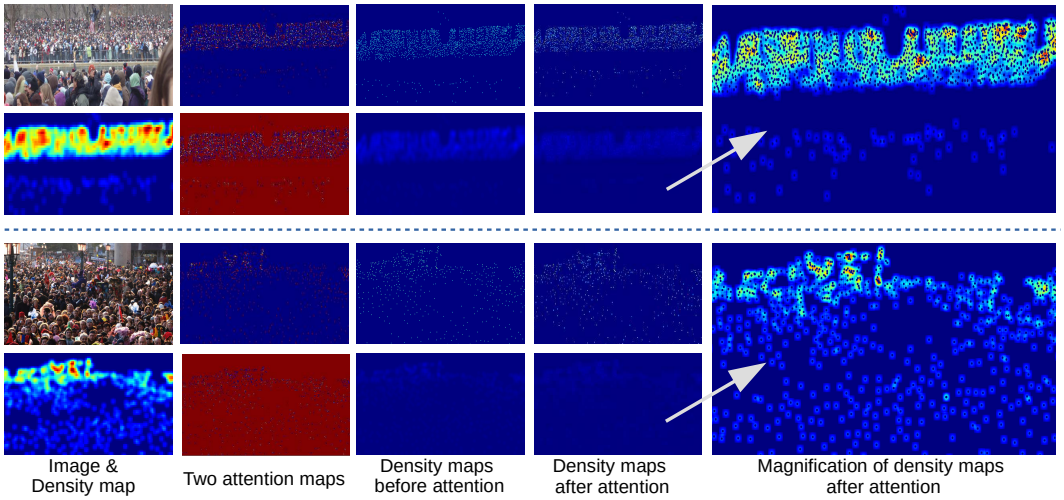

Figure 7. Attention maps, blurred maps before and after attention. The right column shows a magnification of the attention density, which is scaled for better visualization. Only two attention maps are shown due to space limitations.

Table 9. Experimental results on UCF-QNRF.

Method	MAE↓	MSE↓
Multi-sources [12]	315	508
MCNN [42]	277	426
Encoder-decoder [2]	277	426
Cascaded-MTL [32]	252	514
Switching-CNN [29]	228	445
Resnet101 [9]	190	277
Densenet201 [10]	163	226
Composition Loss [13]	132	191
SFCN†[37]	115	192
CSRNet [20]	148	234
Ours (refinement)	111	189
Ours (generation)	**101**	**176**

Table 10. Experimental results on WorldExpo. MAE is the evaluation metric.

Method	S1	S2	S3	S4	S5	Avg.
Cross-scene [41]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [42]	3.4	20.6	12.9	12.0	8.1	11.6
SwitchingCNN [29]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [33]	2.9	14.7	10.5	10.4	5.8	8.86
CNN-pixel [15]	2.9	18.6	14.1	24.6	6.9	13.4
Body structure [11]	4.1	21.7	11.9	11.0	3.5	10.5
DecideNet [21]	**2.0**	13.1	8.9	17.4	4.8	9.2
Top-Down [28]	2.7	23.4	10.7	17.6	3.3	11.5
CSRNet [20]	2.9	**11.5**	8.6	16.6	3.4	8.6
IG-CNN [1]	2.6	16.1	10.2	20.2	7.6	11.3
ic-CNN [26]	17.0	12.3	9.2	**8.1**	4.7	10.3
SANet [3]	2.6	13.2	9.0	13.3	3.0	8.2
SpatialAware [22]	2.6	11.8	10.3	10.4	3.7	**7.76**
ImagePyramid [14]	2.5	16.5	12.2	20.5	**2.9**	10.9
Ours (refinement)	3.8	14.5	11.7	17.9	3.5	10.3
Ours (generation)	4.0	18.1	**7.2**	12.3	5.7	9.5

Similarly, On ShanghaiTech B, both of our frameworks achieve better performance than the baseline CSRNet. The proposed generation method achieves the best MAE (beating SANet), while MSE is similar to SANet.

UCF-QNRF is the most challenging and latest crowd counting dataset, and our proposed methods achieve the best performance on both MAE and MSE. Finally, WorldExpo tests cross-scene performance. On this dataset, the proposed method achieves best performance on test scene 3. However, since the testing image and the training images are from different videos, no method achieves best performance across all scenes.

In summary, these experiments demonstrate that the proposed density map refinement and generation frameworks can produce learnable density map representations that improve counting performance, especially on large datasets such as ShanghaiTech A/B and UCF-QNRF.

6. Conclusion

In this paper, we propose a density map refinement framework to improve the performance of crowd counting by training the counter with refined density maps. By jointly training of counter and refiner, the counting performance is improved. We also propose an adaptive density map generator, which directly uses the dot map as input to generate a density map for training the counter. This end-to-end framework jointly trains the density map generator and counter, and removes the need for hand-specifying the density map as an intermediate representation. The proposed method achieves state-of-the-art performance on 3 popular datasets.

Acknowledgment

This work was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. [T32-101/15-R] and CityU 11212518), and by a Strategic Research Grant from City University of Hong Kong (Project No. 7004887).

References

[1] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3626, 2018. 2, 7, 8

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 8

[3] 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. 2, 7, 8

[4] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008. 1, 2

[5] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *International Conference on Computer Vision*, pages 545–551, 2009. 2

[6] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *British Machine Vision Conference*, volume 1, page 3, 2012. 1

[7] Junyu Gao, Wei Lin, Bin Zhao, Dong Wang, Chenyu Gao, and Jun Wen. C^ 3 framework: An open-source pytorch code for crowd counting. *arXiv preprint arXiv:1907.02724*, 2019. 1

[8] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 1

[9] 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. 8

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 8

[11] Siyu Huang, Xi Li, Zhongfei Zhang, Fei Wu, Shenghua Gao, Rongrong Ji, and Junwei Han. Body structure aware deep crowd counting. *IEEE Trans. Image Processing*, 27(3):1049–1059, 2018. 8

[12] 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. 2, 8

[13] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018. 2, 4, 8

[14] Di Kang and Antoni B. Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *British Machine Vision Conference*, page 89, 2018. 1, 2, 3, 5, 7, 8

[15] Di Kang, Zheng Ma, and Antoni B Chan. Beyond counting: comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2, 8

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[17] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 878–885, 2005. 1

[18] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010. 1, 2

[19] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *International Conference on Pattern Recognition*, pages 1–4, 2008. 2

[20] 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. 3, 5, 7, 8

[21] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2018. 7, 8

[22] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 849–855, 2018. 2, 7, 8

[23] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 7

[24] Zheng Ma, Lei Yu, and Antoni B Chan. Small instance detection by integer programming on object density maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3689–3697, 2015. 2

[25] Mark Marsden, Kevin McGuinness, Suzanne Little, and Noel E. O'Connor. Fully convolutional crowd counting on highly congested scenes. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 27–33, 2017. 7

[26] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *European Conference on Computer Vision*, pages 278–293, 2018. 2, 7, 8

[27] Weihong Ren, Di Kang, Yandong Tang, and Antoni B Chan. Fusing crowd density maps and visual object trackers for people tracking in crowd scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5353–5362, 2018. 2

[28] Deepak Babu Sam and R. Venkatesh Babu. Top-down feedback for crowd counting convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7323–7330, 2018. 2, 7, 8

[29] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4031–4039, 2017. 2, 7, 8

[30] 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. 7

[31] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5382–5390, 2018. 2

[32] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6, 2017. 7, 8

[33] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *IEEE International Conference on Computer Vision*, pages 1879–1888, 2017. 1, 2, 7, 8

[34] Vishwanath A Sindagi and Vishal M Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16, 2018. 2

[35] Elad Walach and Lior Wolf. Learning to count with cnn boosting. In *European Conference on Computer Vision*, pages 660–676. Springer, 2016. 2

[36] Jia Wan, Wenhan Luo, Baoyuan Wu, Antoni B Chan, and Wei Liu. Residual regression and semantic prior for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[37] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019. 3, 5, 7, 8

[38] Ze Wang, Zehao Xiao, Kai Xie, Qiang Qiu, Xiantong Zhen, and Xianbin Cao. In defense of single-column networks for crowd counting. In *British Machine Vision Conference*, page 78, 2018. 7

[39] Bo Wu and Ramakant Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 90–97, 2005. 1

[40] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. Spatiotemporal modeling for crowd counting in videos. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5161–5169, 2017. 2

[41] Cong Zhang, Hongsheng Li, X. Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015. 1, 2, 3, 4, 7, 8

[42] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016. 1, 2, 3, 4, 5, 7, 8