2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)

Deep Crowd Counting In Congested Scenes Through Refine Modules

1st Tong Li*†, 2nd Chuan Wang*†, 3rd Xiaochun Cao*‡†
*SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences
†School of Cyber Security, University of Chinese Academy of Sciences
‡Cyberspace Security Research Center, Peng Cheng Laboratory

***Abstract*—Crowd counting, which aims to predict the number of persons in a highly congested scene, has been widely explored and can be used in many applications like video surveillance, pedestrian flow, etc. The severe mutual occlusion among person, the large perspective distortion and the scale variations always hinder an accurate estimation. Although existing approaches have made much progress, there still has room for improvement. The drawbacks of existing methods are 2-fold: (1)the scale information, which is an important factor for crowd counting, is always insufficiently explored and thus cannot bring well-estimated results; (2)using a unified framework for the whole image may result to a rough estimation in subregions, and thus leads to inaccurate estimation. Motivated by this, we propose a new method to address these problems. We first construct a crowd-specific and scale-aware convolutional neural network, which considers crowd scale variations and integrates multi-scale feature representations in the Cross Scale Module (CSM), to produce the initial predicted density map. Then the proposed Local Refine Modules (LRMs) are performed to gradually re-estimate predictions of subregions. We conduct experiments on three crowd counting datasets (the ShanghaiTech dataset, the UCF_CC_50 dataset and the UCSD dataset). Experiments show that our proposed method achieves superior performance compared with the state-of-the-arts. Besides, we conduct experiments on counting vehicles in the TRANCOS dataset and get better results, which proves the generalization ability of the proposed method.**

***Index Terms*—crowd counting, scale variation, density map, mean absolute error**

I. Introduction

Crowd counting, which aims to obtain an accurate number of a highly congested scene, has attracted more attentions. It is widely used in public safety aware places such as pedestrian street, train station, etc. When faced with a crowded scene, which holds complicated environmental conditions like serious occlusion, large perspective distortion, and scale variations on people (as is shown in Fig. 1), it is more challenging to obtain an accurate count. Generally, researches trending for this field usually obtain a density map, which provides weak location information about crowd distribution.

Traditional detection-based methods [1], [2] cast the counting problem as a detection task and use sliding windows to detect body parts. It suffers a lot when facing crowded scenes, where mutual occlusion occurs. Recently, researchers adopt different deep Convolutional Neural Network (CNN) to regress density map [3]–[8], [11]. They address issues in a flow.

Fig. 1. Visualization of some images and density maps on ShanghaiTech [3]. The serious occlusion, large perspective distortion, and scale variations among people throw a great challenge to obtain an accurate count in congested scenes.

Firstly, they design an architecture to capture representative crowd features. Then a whole image or image patches are delivered into the network to obtain the density map. Finally, the predicted density map is employed to calculate the count. To the best of our knowledge, most existing methods concentrate only on the quality of count rather than that of density map. Compared to obtaining a precise count, these methods pay less attention to generate an accurate density map that is used to count. They ignore that only if a more accurate density map is predicted, a more precise count is obtained.

Based on this observation, we focus on two issues which are ignored before. (1) The scale variation of person, which means the number of pixels corresponding to a person varies a lot in the same image captured by same camera, is obvious especially in crowded scenes. However, the scale information, which is an important factor for crowd counting, is always insufficiently explored and thus cannot bring well-estimated results. (2) A unified framework for the whole image may result to a rough estimation in some subregions of density map. For example, the irregular spatial distribution of crowd may lead to the inaccurate estimation. Therefore, for a unified framework, it would be too hard to take care both congested regions and sparse regions at the same time.

In this paper, we propose a novel architecture to estimate an accurate density map. We address the aforementioned two drawbacks by exploring scale information to ease scale variations and deploying refine modules into density map

978-1-7281-4493-1/19/$31.00 ©2019 IEEE
DOI 10.1109/DSAA.2019.00033

estimation to obtain precise local prediction. Specifically, firstly, we use the Density Feature Generator (DFG) to produce crowd-specific representations covering large receptive fields. Then the Cross Scale Module (CSM) is adopted to incorporate the multi-scale context information into the density map estimation as well as outputs an initial density map. Finally we propose the Local Refine Module (LRM) and use multiple LRMs to gradually refine the subregions of the initial density map. Owing to the novel architecture, the three modules work together to generate a more accurate density map and lower Mean Absolute Error (MAE). We conducted extensive experiments on three crowd counting datasets (the ShanghaiTech dataset [3], the UCF_CC_50 [13] dataset, and the UCSD [14] dataset). Fig. 2 shows the overview architecture of proposed method.

The rest of paper is structured as follows: Sec. II concludes previous works on crowd counting as well as their limits. Sec. III introduces our proposed method and network architecture in detail. Sec. IV lists several counting datasets and the analysis of results on these datasets. The ablation studies are also performed to evaluate the effectiveness of proposed components. Finally we conclude our work in Sec. V.

II. RELATED WORKS

Crowd counting in images [3], [8] and videos [34]–[36] is a popular field among researchers. Counting algorithms are roughly divided into three categories: counting by detection [1], [2], counting by regression [17], [18] and counting with CNN [3]–[8], [11].

A. Detection-based Methods

Detection-based approaches [1], [2] of crowd counting sequentially conduct human detection and count the detection results as the final number. They use manually crafted descriptors such as HOG [1] to represent human and use slide windows to detect human parts such as arm, head, etc. Since detection based methods [1], [2] usually detect people directly and then count, they suffer a lot when facing with more crowded scenes, where people may be seriously occluded by each other and even be seen as a bunch of blobs. With this large scale variation among crowds, obtaining general people representations and directly counting by detection cannot offer satisfactory results.

B. Regression-based Methods

When faced with congested scenes, where occlusion and cluster happen a lot, detection-based methods cannot work well. As a result, researchers propose to regress the mapping function between image characteristics and total counts [37]–[39] or object density [17], [18]. Haroon et. al. [40] leverage multiple sources of information such as head detection, texture elements and frequency domain analysis to compute an estimation of the number of individuals presented in an extremely dense crowd. However, approaches that directly regress the total count discard the localization information, which is important to model crowds. Besides, they suffer a lot from background noises. Thus, these group of approaches usually need large amounts of training data and behave poorly when lacking data.

[16] evades the difficulty of the detection task via estimating an image density whose integral over any image region gives the count of objects within that region. Based on the fact that the linear mapping function is hard to obtain, [12] builds the random forest regressors to learn a non-linear function.

C. CNN-based Methods

Recently, density map estimation via CNN has occupied the mainstream. Based on the fact that CNN has the ability to extract powerful representations of images, it becomes extremely useful in image classification [19], [20], object detection [21], [22] as well as segmentation [23], etc. It's a good choice to adopt CNN to conduct crowd counting task [3]–[8], [11]. Researchers project the dotted people location ground truth to a spacial-aware density map and learn to predict the density map through a powerful CNN. The most common practice is to employ single or multiple neural networks to obtain density maps in an end-to-end manner.

The single-column structure tends to extract crowd-specific and representational features while multi-column structure uses different kernel sizes to handle scale variations. CSRNet [11] adopts a single-column structure. It uses the VGG [19] based network as the frontend to extract features and a dilated CNN to enlarge the receptive fields and regress density maps. MCNN [3] is a typical multi-column structure and it addresses scale variations via 3 parallel channels with different kernels to capture different receptive fields. The count is calculated by summing all the values in estimated density map. Switch-CNN [4] consists of 3 CNN regressors with different kernel sizes of convolution layers. A switch classifier is used to automatically link the image to the best CNN regressor. Cascaded-MTL [7] incorporates a classification network into the density map estimation network by learning to classify images to various groups based on the count. CP-CNN [6] tries to generate high-quality density maps by incorporating the global and the local context information, along with density map information into Fusion-CNN [6]. SANet [5] encodes multi-scale features based on multiple scale aggregation modules and decodes the features with a set of transposed convolution layers. ACSCP [9] designs a U-net [10] structured generation network and an adversarial loss is employed to shrink the solution onto a realistic subspace. It also designs a scale-consistency regularizer to enforce that the sum of crowd counts from local patches equals to the overall count of their region union. Haroon et. al. [46] address the crowd counting, density map estimation and localization in dense crowds by inviting a new composition loss. [45] exploits unlabeled data in CNN to improve counting by self-supervised learning to rank.

III. METHODS

The main idea of the proposed framework is that we want to efficiently mine scale information and refine the density map that may have high estimation error.

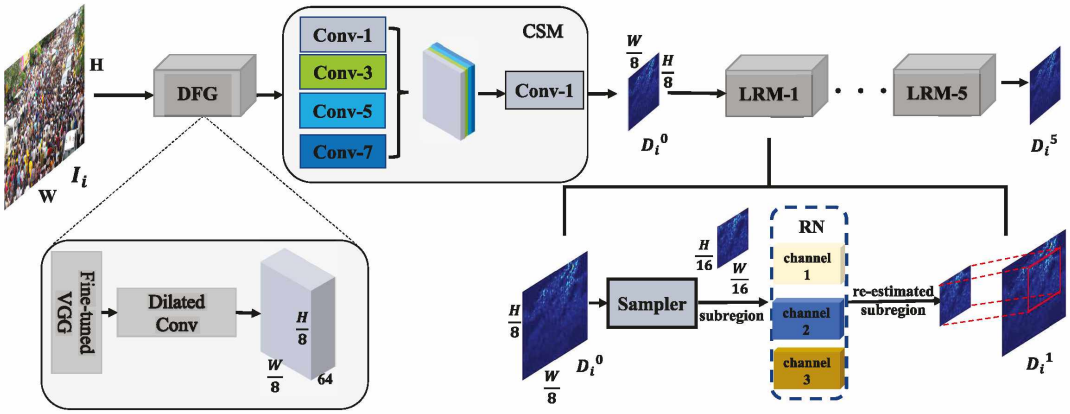


Fig. 2. Overview of the proposed network. Our architecture consists of 3 components. The Density Feature Generator (DFG) provides crowd-specific representations covering large receptive fields. The Cross Scale Module (CSM) provides multi-scale context information about the input image and offers the initial density map. The Local Refine Module (LRM) is proposed to correct density estimation of subregions that may be roughly predicted. In LRM, the Refine Network (RN) is a 3-channels CNN with different convolutional kernel sizes to re-estimate sampled subregion. Then the corresponding subregion in the input density map is replaced by the re-estimated subregion density map. The parameters are shared among LRM.

Motivated by this, we propose our solution to address aforementioned drawbacks in Sec. I. Our proposed architecture consists of 3 components. That is, the Density Feature Generator (DFG) provides crowd-specific representations covering large receptive fields. The Cross Scale Module (CSM) provides multi-scale context information about the input image and offers the initial density map. The Local Refine Module (LRM) is proposed to correct density estimation of subregions that may be roughly predicted.

A. Network Configuration

*1) **Density Feature Generator**:* Researchers are devoted to mine finer features through deeper network, such as ResNet-101 [20], DPN-131 [31]. Our proposed DFG is a simple yet powerful base network intended to generate crowd-specific features. We use the 13 layers of the VGG-16 [19], which contains 10 convolution layers to extract features and 3 max pooling layers to shrink the size to $1/8$ of original resolution. ReLU activation functions are used to increase the non-linear representation capability.

Besides, 6 dilated convolution layers [24] are placed behind VGG-16 [19] layers in order to increase receptive fields as well as capture more contextual information. The dilated convolution [24] is first applied in semantic segmentation [25], [26]. It allows exponential increase in the field of view without decrease of resolution caused by pooling operations. A 2-D dilated convolution can be expressed as:

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

where $x(m,n)$ represents pixel location, $y(m,n)$ represents output result, $w(i,j)$ stands for the filter parameters and M, N is the width and height of x, respectively. Compared to the normal 2-D convolution operation, the dilated convolution introduces a parameter r, which is named as the dilation rate. When $r = 1$, it equals to a normal 2-D convolution. The detail architecture of DFG is shown in Table. I. In our network, DFG is used to output the 64 channels crowd-specific representations of the image for later use.

*2) **Cross Scale Module**:* Previous state-of-the-art approaches usually design a Deep Neural Network (DNN) to conduct crowd counting. Due to the scale variations among real-world crowd scenes, it would be tough for these methods to adapt to both crowded and sparse scenes. One of the possible explanations is that the scale information, which is an important factor for crowd counting, is always insufficiently explored and thus cannot bring well-estimated results. Inspired by the success of the Inception Network [27]–[30] in image classification, we address this problem by adding a CSM behind DFG, which aims to capture multi-scale context information. It is composed of 4 convolutional branches holding different kernel sizes. The first branch is convolved with a kernel having 1*1 size, which is designed to preserve information from the former layer. The other three branches use kernels having sizes of 3*3, 5*5, 7*7,respectively, to capture response from different scales. Then, these three branches are followed by a convolution layer with kernel size 1*1 to reduce the dimension of the channel to 1. Finally, the four feature maps are then concatenated and then convolved with a 1*1 kernel to produce the initial density map. Fig. 3 shows the architecture.

*3) **Local Refine Module**:* Another drawback of previous state-of-the-art methods is that the estimated density map may be too rough in certain subregions and often cause large estimation errors compared to sparse regions. The reason is that a unified framework cannot take care both crowded regions and sparse regions at the same time, they tend to over-estimate the count of sparse regions. We propose to address the problem

TABLE I

THE ARCHITECTURE OF DFG. THE CONVOLUTION LAYER IS EXPRESSED AS CONV(KERNEL SIZE, NUMBER OF FILTERS, DILATION RATE).

	Convolution layer	Feature map size
Density Feature Generator	conv(3, 64, 1) conv(3, 64, 1)	$H \times W \times 64$
	Max Pooling	$\downarrow 2$
	conv(3, 128, 1) conv(3, 128, 1)	$\frac{H}{2} \times \frac{W}{2}$
	Max Pooling	$\downarrow 2$
	conv(3, 256, 1) conv(3, 256, 1) conv(3, 256, 1)	$\frac{H}{4} \times \frac{W}{4} \times 128$
	Max Pooling	$\downarrow 2$
	conv(3, 512, 1) conv(3, 512, 1) conv(3, 512, 1)	$\frac{H}{8} \times \frac{W}{8} \times 256$
	Dilated Convolution layer conv(3, 512, 2) conv(3, 512, 2) conv(3, 512, 2)	$\frac{H}{8} \times \frac{W}{8} \times 512$
	conv(3, 256, 2)	$\frac{H}{8} \times \frac{W}{8} \times 256$
	conv(3, 128, 2)	$\frac{H}{8} \times \frac{W}{8} \times 128$
	conv(3, 64, 2)	$\frac{H}{8} \times \frac{W}{8} \times 64$

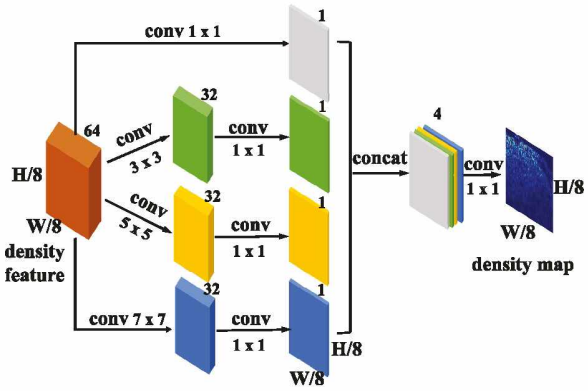


Fig. 3. Overview of the Cross Scale Module (CSM). It is composed of 4 convolutional branches holding different kernel sizes to capture the responses from different scales. All filters are padded to ensure the size of output will not change.

as follows: firstly, we randomly sample a one forth subregion of the estimated density map from CSM, then the sampled subregion is refined through a proposed Refine Network (RN). The RN is composed of 3 branches, each of which consists of 6 convolutional layers with different kernel sizes. The outputs of 3 branches are concatenated to produce the re-estimation of the

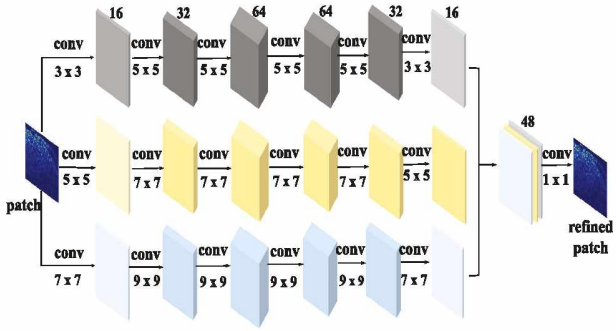


Fig. 4. Overview of the Refine Network (RN). The RN is proposed to correct the density estimation of subregions that may be roughly predicted. It is a 3-branches CNN with different convolutional kernel sizes to re-estimate the sampled subregion. All filters are padded to ensure the same size as the former.

subregion. Then, the difference between the re-estimation of the ground truth is fed into the RN to refine the re-estimation. The overview of the RN is shown in Fig. 4. Here all filters are also padded to ensure the same size as former. After that, the refined density map of the subregion takes place of that in the previous density map produced by CSM.

B. Training With DFG And CSM

The DFG and the CSM are jointly trained in an end-to-end manner with the following Euclidean loss:

$$Loss = \sum_{i=1}^{N} \|\mathbf{D}_i^{GT} - \mathbf{D}_i^0\|^2, \tag{2}$$

where N is number of pictures, $\mathbf{D}_i^0$ is the initial density map estimated by CSM and $\mathbf{D}_i^{GT}$ is ground truth for image I_i. We use the Pytorch [32] framework and the Stochastic Gradient Descent (SGD) as an optimizer with learning rate varying in different datasets(i.e. 10^{-6} on the ShanghaiTech [3] and the UCF_CC_50 [13], 10^{-5} on the UCSD [14] and the TRANCOS [15]). Since the image resolution differs, we set the batch size to 1 and use the whole image as well as image patches during the training stage. The image patches are randomly sampled from the whole image with $1/4$ of the original size. The estimation is performed on the whole image in the testing stage.

C. Training with LRM

After obtaining the initial density map generated by CSM, we froze the parameters of DFG and CSM and train LRM only.

As the backward traditional pixel-wise Euclidean loss depends on the magnitude of deviation of the certain pixel, it tends to incentivize a blur when it confronts sharp edges and outliers [9]. We combine the Structural Similarity In Image (SSIM) loss and Euclidean loss together to overcome this issue. The SSIM index is proposed to measure the local consistency of the generated density map and its ground truth,

which is widely used in the image quality assessment task. The value of SSIM varies from -1 to 1 and equals to 1 only if the two images are identical while 0 if no structural similarity. It evaluates the image quality from three statistics: mean, variance and covariance. It is expressed as

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}, \quad (3)$$

where μ_X and σ_X^2 are the mean and the variance estimation of X, and σ_{XY} represents the covariance estimation. c_1 and c_2 are small constants to avoid division by zero. Then, we define the SSIM loss for a pair as:

$$L_{ssim}(\mathbf{D}_i^{GT}, \mathbf{D}_i) = -\lg \frac{1 - SSIM(\mathbf{D}_i^{GT}, \mathbf{D}_i)}{2}. \quad (4)$$

Finally, the LRM is trained with:

$$Loss = \sum_{i=1}^{N}\sum_{k=1}^{K} \alpha * \|\mathbf{D}_i^{GT} - \mathbf{D}_i^k\|^2 + \beta L_{ssim}(\mathbf{D}_i^{GT}, \mathbf{D}_i^k), \quad (5)$$

where $\mathbf{D}_i^k$ is the refined density map obtained by k-th LRM, K is the number of employed LRMs, α and β are constant values. We set α to 0.2 and β to 0.01, $K = 5$ on ShanghaiTech [3] and UCF_CC_50, $K = 1$ on UCSD [14] and TRANCOS [15]. The parameters are shared by K LRMs. During each step, a subregion is randomly sampled from previous density map and then refined by RN. After that, the refined subregion of density map takes the place of the previous one. The visualization of the re-estimated density map by each LRM is shown in Fig. 5. With the efforts of these procedures, our estimated density map becomes accurate gradually.

IV. EXPERIMENTS

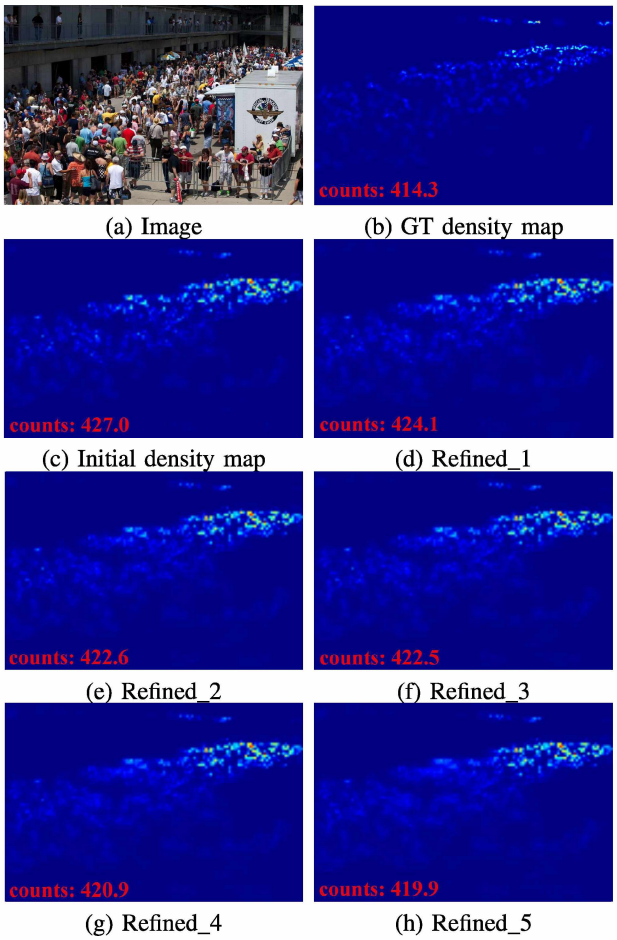
In this section, we make experiments to compare with the state-of-the-art methods on three widely explored crowd counting datasets: the ShanghaiTech [3], the UCF_CC_50 [13] and the UCSD [14] datasets. We also conduct experiments on a vehicle counting dataset, the TRANCOS [15], to illustrate the generalization ability of the proposed method on counting other objects. Results show that our proposed method achieves 5.4% lower Mean Absolute Error(MAE) on the ShanghaiTech [3] dataset, 20% lower MAE on the UCF_CC_50 [13] dataset, compared to state-of-the-art methods. We get a 18.3% lower MAE on the TRANCOS [15] dataset. The ablation study is also performed to evaluate the effectiveness of our proposed components.

A. Evaluation Metrics

To evaluate the accuracy of the estimated density map, we use the following metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE), which are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \|C_i^{GT} - \hat{C}_i\|, \quad (6)$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|C_i^{GT} - \hat{C}_i\|^2}, \quad (7)$$



(a) Image (b) GT density map (c) Initial density map (d) Refined_1 (e) Refined_2 (f) Refined_3 (g) Refined_4 (h) Refined_5

Fig. 5. Visualization of re-estimated density map and corresponding counts by each LRM on ShanghaiTech [3].

where $\hat{C}_i$ represents the predicted count from the estimated density map $\mathbf{D}_i^K$ of image I_i and C_i^{GT} represents the corresponding ground truth count.

B. Ground Truth Generation

Since the counting network usually estimates the density map, it's of vital importance to generate the ground truth density map for each image. Considering that the label of a crowd image is made up of the coordinates of head centers, here we use a delta function $\delta(\mathbf{x} - \mathbf{x}_i)$ to represent a head centered at $\mathbf{x}_i$. Thus a labeled image with M heads inside at $\mathbf{x}$ is expressed as:

$$H(\mathbf{x}) = \sum_{i=1}^{M} \delta(\mathbf{x} - \mathbf{x}_i). \quad (8)$$

Following with the previous method [3], we convert the coordinate annotations to the corresponding density map by

TABLE II
THE PARAMETER SETTINGS FOR DIFFERENT DATASETS.

Dataset	Generating method	K
ShanghaiTech Part_A [3]	Geometry-adaptive kernels	5
ShanghaiTech Part_B [3]	Geometry-adaptive kernels	5
UCF_CC_50 [13]	Geometry-adaptive kernels	5
UCSD [14]	Fixed kernel, σ = 3	1
TRANCOS [15]	Fixed kernel, σ = 3	1

TABLE III
STATISTICS OF DATASETS

Dataset	Number of images	Average Resolution	Count Statistics			
			Total	Min	Max	Avg
ShanghaiTech Part_A [3]	482	589 x 868	241,677	33	3139	501
ShanghaiTech Part_B [3]	716	768 x 1024	88,488	9	578	123
UCF_CC_50 [13]	50	2101 x 2888	63,974	94	4543	1279
UCSD [14]	2000	158 x 238	49,885	11	46	25

convolving with a Gaussian kernel G_{σ_i} [33]. The ground truth density map at $\mathbf{x}$ is calculated through:

$$\mathbf{D}^{GT}(\mathbf{x}) = \sum_{i=1}^{M} \delta(\mathbf{x} - \mathbf{x}_i) * G_{\sigma_i}, \tag{9}$$

where $\sigma_i = \beta \overline{d}^i$ and $\overline{d}^i$ is the average distance to its k nearest neighbors. Following the previous work [3], we set k to 3 and β to 0.3. For each coordinate located at $\mathbf{x}$, it is convolved with the Gaussian kernel with the standard deviation σ_i that is proportional to $\overline{d}^i$. By far, the pixel associated with $\mathbf{x}_i$ corresponds to an area with a radius proportionated to $\overline{d}^i$. In order to work well with sparser datasets like the UCSD [14], we fix the kernel for all image. The detailed setting of σ is shown in Table. II. Once we obtain the density map, the crowd count is obtained through

$$\hat{C}_i = \sum_{i=1}^{h} \sum_{j=1}^{w} \mathbf{D}(i,j), \tag{10}$$

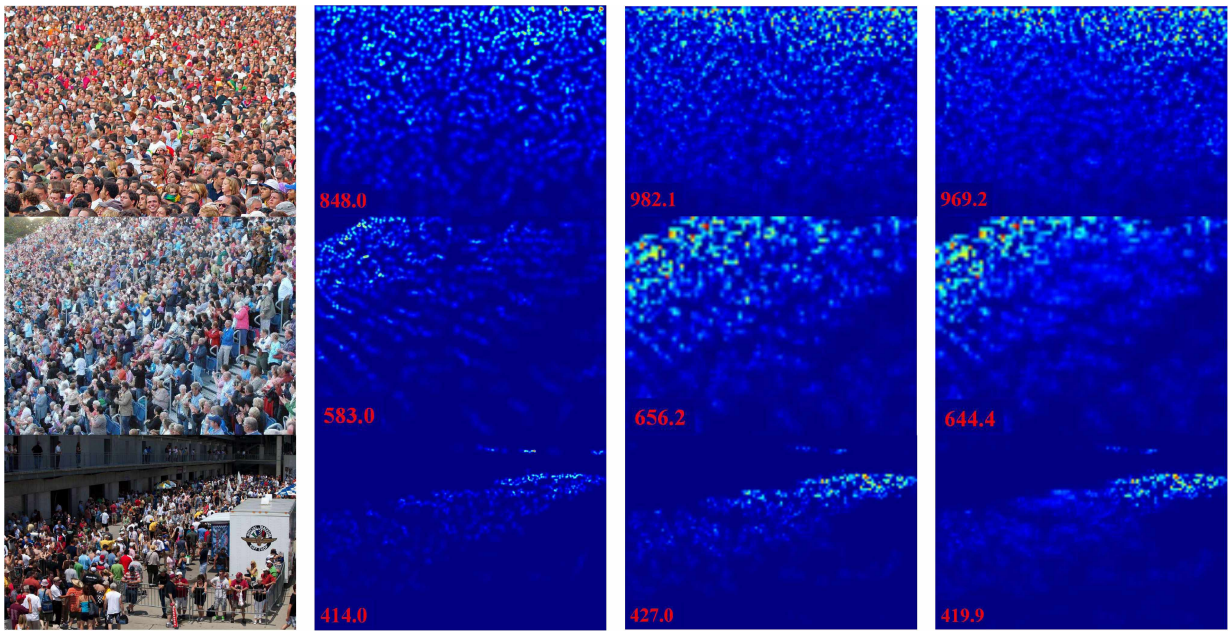
where h and w represent the height and the width of density map, respectively.

C. Data Augmentation

Since some datasets have limited number of images(e.g. the UCF_CC_50 [13] contains only forty images for training and ten images for testing), it is necessary to use some tricks to enrich the training data. During the training stage, we use the whole image and patches that randomly sampled with $1/4$ of original resolution. A random flip operation is made to double the training set. During the inference stage, the whole image is delivered into the network.

D. Datasets And Settings

- **ShanghaiTech** The ShanghaiTech [3] dataset, which contains 1198 annotated images with a total of 330,165 people with centers of head annotations, is a widespread dataset in the crowd counting fields. It consists of two parts: part A with 300 images for training and 182 images for testing; part B with 400 images for training and 316 images for testing. We use the Stochastic Gradient Descend (SGD) as the optimizer with an initial learning rate 10^{-6}. Data augmentation is used to go a step further. We deploy 5 LRMs to refine the initinal predicted density map step by step and the parameters are shared among LRMs. By the way, due to the existence of random sampling in the LRM, there exists slight difference when we perform the inference at each time. As a result, to reduce the differences and make the results more convincing, we repeat the testing step 50 times and take the average counts as the final reports.
- **UCF_CC_50** The UCF_CC_50 [13] dataset contains 50 images collected from the Internet. It is a very challenging dataset because the head counts of each image vary from 94 to 4543, with an average number of 1279 head counts per image. The number of overall head annotations is 63974. A 5-fold cross-validation step is performed on this dataset. Considering that the test set is relative small and the random operation is introduced in our method, we infer 100 times per image and take an average MAE and MSE as the final results. We deploy 5 LRMs similar to ShanghaiTech [3].
- **UCSD** The UCSD [14] dataset contains 2000 images captured by surveillance cameras. It is sparser when compared with the UCF_CC_50 [13] and the ShanghaiTech [3]. The count of each image varies from 11 to 46. We conduct experiments on this dataset to prove the strong generalization capability of the proposed method. Since the dataset is sparser than the ShanghaiTech [3] and the UCF_CC_50 [13], we deploy only 1 LRM to re-estimate the density map. Based on the fact that the image resolution is relative small (only 158x238), it would be difficult to directly deliver the images into our network, which contains 3 max pooling layers and outputs a 1/8 size feature map compared with the original input (as listed in Table. I). As a result, we resize the images into 632x952 by using the bilinear interpolation before delivering into our network. In addition, we mask all images and the corresponding ground truth maps by the Region-Of-Interest (ROI) map. For a quick comparison of these datasets, please refer to Table. III.
- **TRANCOS** Apart from the aforementioned three crowd counting datasets, we evaluate our method on a vehicle counting dataset named TRANCOS [15] in order to demonstrate the generalization capability and practical applications. The TRANCOS [15] dataset contains 1224 traffic jam images captured by surveillance cameras covering different scenarios. It has 46796 annotated vehicles with ROI map provided. We deploy 1 LRM similar to the UCSD [14] dataset. Following the previous work [15], we use the Grid Average Mean Absolute Error(GAME) as the measurement. The GAME metric is introduced to provide a more accurate evaluation. Different from the



(a) Image (b) Ground truth (c) Initial density map (d) Refined density map

Fig. 6. Visualization of some images and their ground truth and predicted density maps in ShanghaiTech [3]. First column: original images. Second column: ground truth. Third column: initial density map. Forth column: refined density map.

TABLE IV
COMPARISON BETWEEN CURRENT METHODS AND OUR PROPOSED METHOD ON THE SHANGHAITECH DATASET [3].

Methods	PartA		PartB	
	MAE	MSE	MAE	MSE
MCNN [3]	110.2	173.2	26.4	41.3
Cascaded-MTL [7]	101.3	152.4	20.0	31.1
Switch-CNN [4]	90.4	135.0	21.6	33.4
ACSCP [9]	75.7	102.7	17.2	27.4
CP-CNN [6]	73.6	106.4	20.1	30.1
CSRNet [11]	68.2	115.0	10.6	16.0
SANet [5]	67.0	104.5	**8.4**	**13.6**
Liu et al. [45]	72.0	106.6	13.7	21.4
ours	**64.4**	**97.6**	10.1	15.6

TABLE V
COMPARISON WITH CURRENT STATE-OF-THE-ART METHODS ON THE UCF_CC_50 [13] AND THE UCSD [14] DATASETS.

Method	UCF_CC_50 [13]		UCSD [14]	
	MAE	MSE	MAE	MSE
MCNN [3]	377.6	509.1	1.07	1.35
Cascaded-MTL [7]	322.8	341.4	-	-
Switch-CNN [4]	318.1	439.2	1.62	2.10
ACSCP [9]	291.0	404.6	1.04	1.35
CP-CNN [6]	295.8	320.9	-	-
CSRNet [11]	266.1	397.5	1.16	1.47
SANet [5]	258.4	334.9	**1.02**	**1.29**
Liu et al. [45]	279.6	397.6	1.17	1.55
ours	**206.7**	**276.8**	1.10	1.39

MAE, the GAME metric takes object counts as well as object locations into consideration. For a specific level L, it subdivides the image into 4^L (L=0,1,2,3, respectively) non-overlapping regions and calculates the MAE in each region. It is defined as follows,

$$GAME(L) = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{4^L}\|\hat{C}_i^l - C_i^l\|, \qquad (11)$$

where $\hat{C}_i^l$ is the estimated count in a region l of image i, C_i^l is the corresponding ground truth for the same region of image i. When L = 0, it equals to the MAE. The GAME is a restrict evaluation metric for that the higher L, the more subregions are divided.

E. Quantitative Results And Analysis

Table. IV shows the comparison with previous state-of-the-art methods on the ShanghaiTech [3], which proves the effectiveness of our new architecture on the crowd counting task. As it is shown in Table. IV, we get a 64.4 of MAE and 97.6 of MSE on part A, which both outperform the current state-of-the-art method SANet [5] by 5.4% lower MAE and 6.6% lower MSE. Fig. 6 shows some visualization of images and their corresponding predictions of our method on the ShanghaiTech [3].

Compared to the ShanghaiTech [3] dataset, the UCF_CC_50 [13] is more challenging. Table. V shows the results, from which we observe that our proposed method surpasses the current state-of-the-art method SANet [5] with large margins

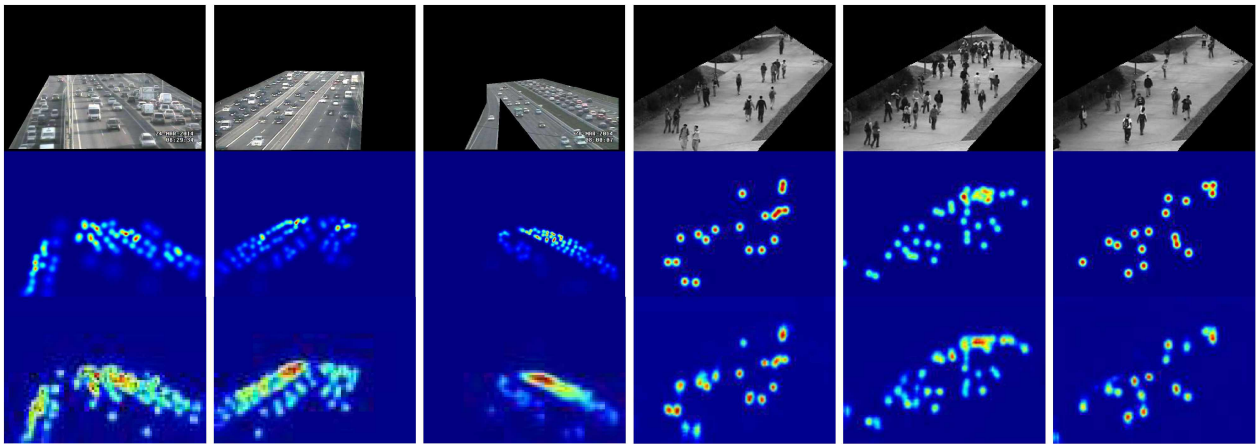


Fig. 7. The visualization of some masked images and density maps on the TRANCOS [15] and the UCSD [14] datasets. First row: examples from TRANCOS [15] and UCSD [14]. Second row: the ground truth density maps (60.8, 61.8, 67,0, 19.0, 40.6, 17.0). Third row: the predicted density maps (65.7, 70.0, 70.3, 18.9, 41.0, 17.3).

TABLE VI
RESULTS ON TRANCOS [15] DATASET.

Method	GAME Metric			
	L = 0	L = 1	L = 2	L = 3
Fiaschi [41] et al.	17.77	20.14	23.65	25.99
Lempitsky et al. [42]	13.76	16.72	20.72	24.36
Hydra-3s [43]	10.99	13.75	16.69	19.32
FCH-HA [44]	4.21	-	-	-
CSRNet [11]	3.56	**5.49**	**8.57**	15.04
ours	**2.91**	6.23	11.51	**14.91**

(i.e. 20% lower MAE and 17.3% lower MSE). Since the UCF_CC_50 [13] dataset has large scale variations among people, our proposed CSM handles these kind of scale variations easily. Further more, the UCF_CC_50 [13] dataset has more congested areas, which result to higher probability of estimating high estimation errors on those dense subregions by previous methods. Different from them, our proposed LRM is intended to address this problem and has positive impacts on decreasing MAE of areas with high estimation errors. In comparison, SANet [5] adopts several scale aggregation modules to encode features, which is not powerful than the DFG module proposed in this paper. Besides, SANet [5] lacks refinements of the density map, which is our key point since the UCF_CC_50 dataset has more congested regions.

As is illustrated in Table. V, when faces with sparse scenes in UCSD [14], we have a competitive results compared to SANet [5]. It is obvious that our proposed method works well both on congested scenes and sparse scenes. That is, it is robust to density variations.

Table. VI lists our results on the TRANCOS [15] dataset. We surpass all previous methods when $L = 0$ and $L = 3$. It is glad to see that our method has a good generalization capability of transferring to count other objects such as vehicles.

F. Ablation Studies

We conduct the experiments on two datasets (the ShanghaiTech [3] part A and the UCF_CC_50 [13]) to evaluate the effectiveness of our proposed modules. Considering the DFG aims to generate density features of the input image, we add an output layer consisting of one convolution layer with the kernel size 1 to generate the density map. After obtain the baseline, we gradually add other proposed modules, results are shown in Table. VII.

TABLE VII
THE ABLATION STUDIES OF PROPOSED MODULES ON THE UCF_CC_50 [13] AND THE SHANGHAITECH A [3] DATASETS.

Method	UCF_CC_50 [13]		ShanghaiTech [3]	
	MAE	MSE	MAE	MSE
baseline	239.5	321.4	67.0	105.0
+CSM	214.2	307.9	65.7	100.7
+CSM+LRM	**206.7**	**276.8**	**64.4**	**97.6**

- **The effectiveness of the DFG.** The DFG is a simple yet powerful network structure that intended to generate crowd-specific features. Its frontend is an Imagenet-pretrained thirteen-layer-VGG-16 [19] and backend is composed of 6 dilated convolution layers that aim to increase receptive fields as well as capture more contextual information. We obtain comparable MAE and MSE compared with current methods [5] [11]. As it is shown in Table. VII, the DFG is a strong extractor to generate density features, which are of vital importance to generate density map with high quality.
- **Effectiveness of CSM.** The scale information, which is an important factor for crowd counting, is always insufficiently explored and thus cannot bring well-estimated results. So the CSM is proposed to capture scale variations among crowds. From Table. VII, we observe that after

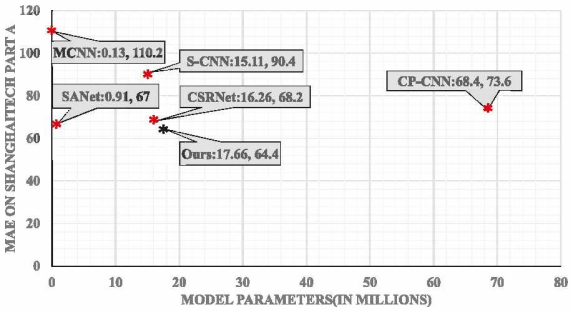


Fig. 8. Contrast of model parameters and the MAE on the ShanghaiTech [3] part A. Our architecture makes a deal with the two targets and surpasses the state-of-the-art methods without much cost of parameters.

adapting the CSM to the network, we achieve superior results on the UCF_CC_50 [13] dataset and 1.3 lower MAE on the ShanghaiTech part A [3] (as is shown in Table. VII). This is a strong support for the notion, that convolution layers with different kernel size have the ability to ease scale variations.

- **Effectiveness of LRM.** Considering that the estimated density map may be too rough in certain subregions and thus cause huge estimation errors compared to sparse regions, we deploy the LRM into network. After applied the LRM, we obtain another 7.5 lower MAE on the UCF_CC_50 [13], 1.3 lower MAE on the ShanghaiTech part A [3]. The superior results prove that our proposed LRM is useful in refining density maps. In Fig. 5, we also provide a set of images to demonstrate the effects after each refinement. We observe that after each refinement, the corresponding count is closer to the ground truth.
- **Effectiveness of our architecture.** As it is shown in Fig. 8, we make an "accuracy-size" trade-off. We try to steer a middle course between the quality of the generated density map and the parameters that costs. More specifically, the MCNN [3] achieves the state-of-the-art before 2017 at the cost of only 0.13 million parameters, which is quite an effective framework. However, it is not satisfying when facing congested scenes. The model size determines its ability to process complicated scenes to a certain extent. Another extreme is that the CP-CNN [6] designs a very complicated network which is composed of four sub-networks, i.e. the Global Context Estimator (GCE), the Local Context Estimator (LCE), the Density Map Estimator (DMP) and the Fusion-CNN (F-CNN). It also achieves state-of-the-art at the heavy cost of model size and computer resources. In comparison, our architecture makes a deal with these two targets and surpasses the state-of-the-art methods without much cost of parameters.

V. CONCLUSIONS

In this paper, we address the scale variations and density map estimation in crowd counting task by mining the scale information and refining the density map to re-estimate a more accurate result. Our architecture consists of 3 components: the DFG to generate crowd-specific representations for images, the CSM to incorporate multi-scale context information into density map estimation, and the LRM to refine the density map step by step. Due to the novel architecture, we obtain best results on certain benchmarks.

To the best of our knowledge, the random operations that LRM used may not be the best strategy to select subregions with higher estimation errors that needed to be re-estimated. Therefore, we will replace these operations with more targeted ways in order to focus on really "urgent subregions" in the future.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China (Grant No. 2018YFB0803701). National Natural Science Foundation of China (No. U1636214, 61861166002, U1803264). Beijing Natural Science Foundation (No. L182057).

REFERENCES

[1] N. Dalal, and Bill Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005, pp. 886-893.
[2] M. Li, Z. Zhang, K. Huang, and T.Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in ICPR, 2008, pp. 1-4.
[3] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-Image crowd counting via multi-column convolutional neural network," in CVPR, 2016, pp. 589-597.
[4] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in CVPR, 2017, pp. 4031-4039.
[5] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in ECCV, 2018, pp. 734-750.
[6] V. A. Sindagi, and M. P. Vishal, "Generating high-quality crowd density maps using contextual pyramid cnns," in ICCV, 2017, pp. 1861-1870.
[7] V. A. Sindagi, and M. P. Vishal, "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in AVSS, 2017. pp. 1-6.
[8] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in ACMMM, 2015, pp. 1299-1302.
[9] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in CVPR, 2018, pp. 5245-5254.
[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in MICCAI, 2015, pp. 234-241.
[11] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in CVPR, 2018, pp. 1091-1100.
[12] V.Q.Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in ICCV, 2015, pp. 3253-3261.
[13] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in CVPR, 2013, pp. 2547-2554.
[14] A.B. Chan, Z.S.J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: counting people without people models or tracking," in CVPR, 2008, pp. 1-7.
[15] R. Guerrero-Gmez-Olmedo, B. Torre-Jimnez, R. J. Lpez-Sastre, S. Maldonado-Bascn, and D. Onoro-Rubio, "Extremely overlapping vehicle counting," in IbPRIA, 2015, pp. 423-431.
[16] V. Lempitsky, and A. Zisserman, "Learning to count objects in images," in NIPS, 2010, pp. 1324-1332.

[17] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in CVPR, 2013, pp. 2467-2474.
[18] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in BMVC, 2012, vol. 1, no. 2, p. 3.
[19] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in arXiv preprint, 2014.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in ECCV, 2016, pp. 630-645.
[21] R. Girshick, "Fast r-cnn," in ICCV, 2015, pp. 1440-1448.
[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in NIPS, 2015, pp. 91-99.
[23] K. He, G.Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," in ICCV, 2017, pp. 2961-2969.
[24] F. Yu, and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in arXiv preprint, 2015.
[25] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," TPAMI, vol.40, no.4, pp. 834-848, 2017.
[26] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," in arXiv preprint, 2017.
[27] C. Zegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and A. Rabinovich, "Going deeper with convolutions," in CVPR, 2015, pp. 1-9.
[28] S. Ioffe, and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in arXiv preprint, 2015.
[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in CVPR, 2016, pp. 2818-2826.
[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in AAAI, 2017.
[31] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in NIPS, 2017, pp. 4467-4475.
[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[33] V. Lempitsky, and A. Zisserman, "Learning to count objects in images," in NIPS, 2010, pp. 1324-1332.
[34] G. J. Brostow, and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in CVPR, 2006, pp. 594-601.
[35] W. Ge, and T. C. Robert, "Marked point processes for crowd counting," in CVPR, 2009, pp. 2913-2920.
[36] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," TCSVT, vol. 25, no. 3, pp. 367-386, 2015.
[37] A. B. Chan, and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in ICCV, 2009, pp.545-551.
[38] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in BMVC, 2012, pp. 3.
[39] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in DICTA, 2009, pp. 81-88.
[40] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in CVPR, 2013, pp. 2547-2554.
[41] L. Fiaschi, U. Kthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in ICPR, 2012, pp. 2685-2688.
[42] V. Lempitsky, and A. Zisserman, "Learning to count objects in images," in NIPS, 2010, pp. 1324-1332.
[43] D. Onoro-Rubio, and R. J. Lpez-Sastre, "Towards perspective-free object counting with deep learning," in ECCV, 2016, pp. 615-629.
[44] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Fcn-rlstm: deep spatio-temporal neural networks for vehicle counting in city cameras," in ICCV, 2017, pp. 3667-3676.
[45] X. Liu, V. D. W. Joost, and A. D. Bagdanov, "Exploiting unlabeled data in CNNs by self-supervised learning to Rank," TPAMI, vol. 41, no. 8, pp. 1862-1878, 2019.
[46] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, "Composition loss for counting, density map estimation and localization in dense crowds," in ECCV, 2018, pp. 532-546.