



# Adaptive Mixture Regression Network with Local Counting Map for Crowd Counting

Xiyang Liu<sup>1</sup>, Jie Yang<sup>2</sup>, Wenrui Ding<sup>3</sup>(✉), Tieqiang Wang<sup>4</sup>, Zhijin Wang<sup>2</sup>,  
and Junjun Xiong<sup>2</sup>

<sup>1</sup> School of Electronic and Information Engineering, Beihang University,  
Beijing, China

[xiyangliu@buaa.edu.cn](mailto:xiyangliu@buaa.edu.cn)

<sup>2</sup> Shunfeng Technology (Beijing) Co., Ltd, Beijing, China

[jieyang2@sffmail.sf-express.com](mailto:jieyang2@sffmail.sf-express.com)

<sup>3</sup> Institute of Unmanned Systems, Beihang University, Beijing, China

[ding@buaa.edu.cn](mailto:ding@buaa.edu.cn)

<sup>4</sup> Institute of Automation, Chinese Academy of Sciences, Beijing, China

**Abstract.** The crowd counting task aims at estimating the number of people located in an image or a frame from videos. Existing methods widely adopt density maps as the training targets to optimize the point-to-point loss. While in testing phase, we only focus on the differences between the crowd numbers and the global summation of density maps, which indicate the inconsistency between the training targets and the evaluation criteria. To solve this problem, we introduce a new target, named local counting map (LCM), to obtain more accurate results than density map based approaches. Moreover, we also propose an adaptive mixture regression framework with three modules in a coarse-to-fine manner to further improve the precision of the crowd estimation: scale-aware module (SAM), mixture regression module (MRM) and adaptive soft interval module (ASIM). Specifically, SAM fully utilizes the context and multi-scale information from different convolutional features; MRM and ASIM perform more precise counting regression on local patches of images. Compared with current methods, the proposed method reports better performances on the typical datasets. The source code is available at <https://github.com/xiyang1012/Local-Crowd-Counting>.

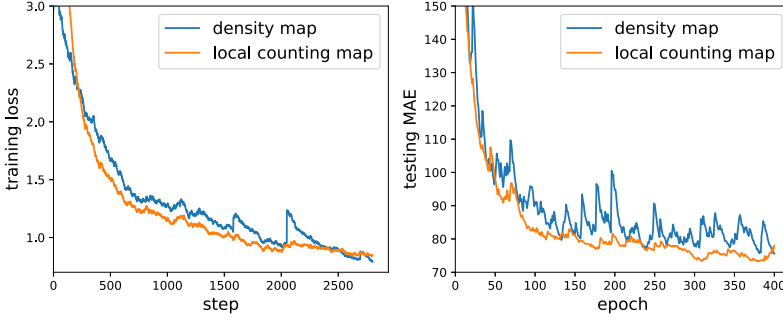
**Keywords:** Crowd counting · Local counting map · Adaptive mixture regression network

---

This work is done when Xiyang Liu is an intern at Shunfeng Technology.

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-58586-0\\_15](https://doi.org/10.1007/978-3-030-58586-0_15)) contains supplementary material, which is available to authorized users.



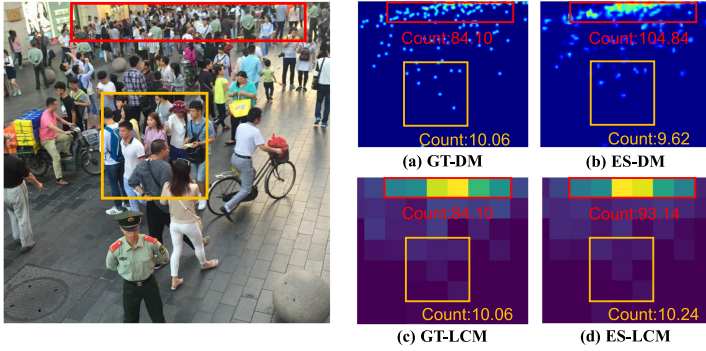
**Fig. 1.** Training loss curves (*left*) and testing loss curves (*right*) between the two networks sharing VGG16 as the backbone with different regression targets, density map and local counting map on ShanghaiTech Part A dataset. The network trained with the local counting map has the lower error and more stable performance on the testing dataset than the one with the density map

## 1 Introduction

The main purpose of visual crowd counting is to estimate the numbers of people from static images or frames. Different from pedestrian detection [12, 15, 18], crowd counting datasets only provide the center points of heads, instead of the precise bounding boxes of bodies. So most of the existing methods draw the density map [11] to calculate crowd number. For example, CSRNet [13] learned a powerful convolutional neural network (CNN) to get the density map with the same size as the input image. Generally, for an input image, the ground truth of its density map is constructed via a Gaussian convolution with a fixed or adaptive kernel on the center points of heads. Finally, the counting result can be represented via the summation of the density map.

In recent years, benefit from the powerful representation learning ability of deep learning, crowd counting researches mainly focus on CNN based methods [1, 3, 20, 25, 36] to generate high-quality density maps. The mean absolute error (MAE) and mean squared error (MSE) are adopted as the evaluation metrics of crowd counting task. However, we observed an inconsistency problem for the density map based methods: the training process minimizes the  $L_1/L_2$  error of the density map, which actually represents a point-to-point loss [6], while the evaluation metrics in the testing stage only focus on the differences between the ground-truth crowd numbers and the overall summation of the density maps. Therefore, the model with minimum training error of the density map does not ensure the optimal counting result when testing.

To draw this issue, we introduce a new learning target, named local counting map (LCM), in which each value represents the crowd number of a local patch rather than the probability value indicating whether has a person or not in the density map. In Sect. 3.1, we prove that LCM is closer to the evaluation metric than the density map through a mathematical inequality deduction. As shown in Fig. 1, LCM markedly alleviates the inconsistency problem brought by



**Fig. 2.** An intuitive comparison between the local counting map (LCM) and the density map (DM) on local areas. LCM has more accurate estimation counts on both dense (the red box) and sparse (the yellow box) populated areas. (*GT-DM*: ground-truth of DM; *ES-DM*: estimation of DM; *GT-LCM*: ground-truth of LCM; *ES-LCM*: estimation of LCM) (Color figure online)

the density map. We also give an intuitive example to illustrate the prediction differences of LCM and density map. As shown in Fig. 2, the red box represents the dense region and the yellow one represents the sparse region. The prediction of density map is not reliable in dense areas, while LCM has more accurate counting results in these regions.

To further improve the counting performance, we propose an adaptive mixture regression framework to give an accurate estimation of crowd numbers in a coarse-to-finer manner. Specifically, our approach mainly includes three modules: 1) scale-aware module (SAM) to fully utilize the context and multi-scale information contained in feature maps from different layers for estimation; 2) mixture regression module (MRM) and 3) adaptive soft interval module (ASIM) to perform precise counting regression on local patches of images.

In summary, the main contributions in this work are in the followings:

- We introduce a new learning target LCM, which alleviates the inconsistency problem between training targets and evaluation criteria, and reports better counting performance compared with the density map.
- We propose an adaptive mixture regression framework in a coarse-to-finer manner, which fully utilizes the context and multi-scale information from different convolutional features and performs more accurate counting regression on local patches.

The rest of the paper is described as follows: Sect. 2 reviews the previous work of crowd counting; Sect. 3 details our method; Sect. 4 presents the experimental results on typical datasets; Sect. 5 concludes the paper.

## 2 Related Work

Recently, CNN based approaches have become the focus of crowd counting researches. According to regression targets, they can be classified into two categories: density estimation based approaches and direct counting regression ones.

### 2.1 Density Estimation Based Approaches

The early work [11] defined the concept of density map and transformed the counting task to estimate the density map of an image. The integral of density map in any image area is equal to the count of people in the area. Afterwards, Zhang *et al.* [35] used CNN to regress both the density map and the global count. It laid the foundation for subsequent works based on CNN methods. To improve performance, some methods aimed at improving network structures. MCNN [36] and Switch-CNN [2] adopted multi-column CNN structures for mapping an image to its density map. CSRNet [13] removed multi-column CNN and used dilated convolution to expand the receptive field. SANet [3] introduced a novel encoder-decoder network to generate high-resolution density maps. HACNN [26] employed attention mechanisms at various CNN layers to selectively enhance the features. PaDNet [29] proposed a novel end-to-end architecture for pan-density crowd counting. Other methods aimed at optimizing the loss function. ADMG [30] produced a learnable density map representation. SPANet [6] put forward MEP loss to find the pixel-level subregion with high discrepancy to the ground truth. Bayesian Loss [17] presented a Bayesian loss to adopt a more reliable supervision on the count expectation at each annotated point.

### 2.2 Direct Counting Regression Approaches

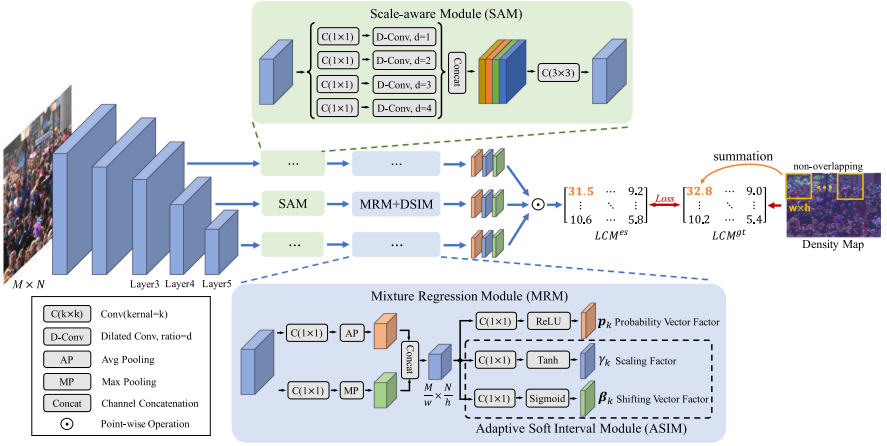
Counting regression approaches directly estimate the global or local counting number of an input image. This idea was first adopted in [5], which proposed a multi-output regressor to estimate the counts of people in spatially local regions for crowd counting. Afterwards, Shang *et al.* [21] made a global estimation for a whole image, and adopted the counting number constraint as a regularization item. Lu *et al.* [16] regressed a local count of the sub-image densely sampled from the input, and merged the normalized local estimations to a global prediction. Paul *et al.* [19] proposed the redundant counting, and it was generated with the square kernel instead of the Gaussian kernel adopted by the density map. Chattopadhyay *et al.* [4] employed a divide and conquer strategy while incorporating context across the scene to adapt the subitizing idea to counting. Stahl *et al.* [28] adopted a local image divisions method to predict global image-level counts without using any form of local annotations. S-DCNet [32] exploited a spatial divide and conquer network that learned from closed set and generalize to open set scenarios.

Though many approaches have been proposed to generate high-resolution density maps or predict global and local counts, the robust crowd counting of

diverse scenes remains hard. Different with previous methods, we firstly introduce a novel regression target, and then adopt an adaptive mixture regression network in a coarse-to-fine manner for better crowd counting.

### 3 Proposed Method

In this section, we first introduce LCM in details and prove its superiority compared with the density map in Sect. 3.1. After that, we describe SAM, MRM and ASIM of the adaptive mixture regression framework in Sects. 3.2, 3.3 and 3.4, respectively. The overview of our framework is shown in Fig. 3.



**Fig. 3.** The overview of our framework mainly including three modules: 1) scale-aware module (SAM), used to enhance multi-scale information of feature maps via multi-column dilated convolution; 2) mixture regression module (MRM) and 3) adaptive soft interval module (ASIM), used to regress feature maps to the probability vector factor  $\mathbf{p}_k$ , the scaling factor  $\gamma_k$  and the shifting vector factors  $\beta_k$  of the  $k$ -th mixture, respectively. We adopt the feature maps of layers 3, 4 and 5 as the inputs of SAM. The local counting map (LCM) is calculated according to parameters  $\{\mathbf{p}_k, \gamma_k, \beta_k\}$  and point-wise operation in Eq. (8). For an input  $M \times N$  image and the  $w \times h$  patch size, the output of the entire framework is a  $\frac{M}{w} \times \frac{N}{h}$  LCM

#### 3.1 Local Counting Map

For a given image containing  $n$  heads, the ground-truth annotation can be described as  $GT(p) = \sum_{i=1}^n \delta(p - p_i)$ , where  $p_i$  is the pixel position of the  $i$ -th head's center point. Generally, the generation of the density map is based on a fixed or adaptive Gaussian kernel  $G_\sigma$ , which is described as  $D(p) = \sum_{i=1}^n \delta(p - p_i) * G_\sigma$ . In this work, we fix the spread parameter  $\sigma$  of the Gaussian kernel as 15.

Each value in LCM represents the crowd number of a local patch, rather than a probability value indicating whether has a person or not in the density map. Because heads may be at the boundary of two patches in the process of regionalizing an image, it's unreasonable to divide people directly. Therefore, we generate LCM by summing the density map patch-by-patch. Then, the crowd number of local patch in the ground-truth LCM is not discrete value, but continuous value calculated based on the density map. The LCM can be described as the result of the non-overlapping sliding convolution operation as follows:

$$LCM = D * \mathbf{1}_{(w,h)}, \quad (1)$$

where  $D$  is the density map,  $\mathbf{1}_{(w,h)}$  is the matrix of ones and  $(w, h)$  is the local patch size.

Next, we explain the reason that LCM can alleviate the inconsistency problem of the density map mathematically. For a test image, we set the  $i$ -th pixel in ground-truth density map as  $g_i$  and the  $i$ -th pixel in estimated density map as  $e_i$ . The total pixels number of the image is  $m$  and the pixels number of the local patch is  $t = w \times h$ . The evaluation criteria of mean absolute error (MAE), the error of LCM (LCME) and the error of density map (DME) can be calculated as follows:

$$MAE = |(e_1 + e_2 + \dots + e_m) - (g_1 + g_2 + \dots + g_m)|, \quad (2)$$

$$LCME = |(e_1 + \dots + e_t) - (g_1 + \dots + g_t)| + \dots + |(e_{m-t} + \dots + e_m) - (g_{m-t} + \dots + g_m)|, \quad (3)$$

$$DME = |e_1 - g_1| + |e_2 - g_2| + \dots + |e_m - g_m|. \quad (4)$$

According to absolute inequality theorem, we can get the relationship among them:

$$MAE \leq LCME \leq DME. \quad (5)$$

When  $t = 1$ , we have  $LCME = DME$ . When  $t = m$ , we get  $LCME = MAE$ .  $LCME$  provides a general form of loss function adopted for crowding counting. No matter what value  $t$  takes,  $LCME$  proves to be a closer bound of  $MAE$  than  $DME$  theoretically.

On the other side, we clarify the advantages of  $LCME$  for training, compared with  $DME$  and  $MAE$ . 1)  $DME$  mainly trains the model to generate probability responses pixel-by-pixel. However, pixel-level position labels generated by a Gaussian kernel may be low-quality and inaccurate for training, due to severe occlusions, large variations of head size, shape and density, etc. There is also a gap between the training loss  $DME$  and the evaluation criteria  $MAE$ . So the model with minimum training  $DME$  does not ensure the optimal counting result when testing with  $MAE$ . 2)  $MAE$  means direct global counting from an entire

image. But global counting is an open-set problem and the crowd number ranges from 0 to  $\infty$ , the MAE optimization makes the regression range greatly uncertain. Meanwhile, global counting would ignore all spatial annotated information, which couldn't provide visual density presentations of the prediction results. 3) LCM provides a more reliable training label than the density map, which discards the inaccurate pixel-level position information of density maps and focuses on the count values of local patches. LCME also lessens the gap between DME and MAE. Therefore, we adopt LCME as the training loss rather than MAE or DME.

### 3.2 Scale-Aware Module

Due to the irregular placement of cameras, the scales of heads in an image are usually very polytropic, which brings great challenge to crowd counting task. To deal with this problem, we propose scale-aware module (SAM) to enhance the multi-scale feature extraction capability of the network. The previous works, such as L2SM [33] and S-DCNet [32], mainly focused on the fusion of feature maps from different CNN layers and acquire multi-scale information through feature pyramid network structure. Different from them, the proposed SAM achieves multi-scale information enhancement only on a single layer feature map and performs this operation at different convolutional layers to bring rich information to subsequent regression modules.

For fair comparisons, we treat VGG16 as the backbone network for CNN-based feature extraction. As shown in Fig. 3, we enhance the feature maps of layers 3, 4 and 5 of the backbone through SAM, respectively. SAM first compresses the channel of feature map via  $1 \times 1$  convolution. Afterwards, the compressed feature map is processed through dilated convolution with different expansion ratios of 1, 2, 3 and 4 to perceive multi-scale features of heads. The extracted multi-scale feature maps are fused via channel-wise concatenation operation and  $3 \times 3$  convolution. The size of final feature map is consistent with the input one.

### 3.3 Mixture Regression Module

Given an testing image, the crowd numbers of different local patches vary a lot, which means great uncertainty on the estimation of local counting. Instead of taking the problem as a hard regression in TasselNet [16], we model the estimation as the probability combination of several intervals. We propose the MRM module to make the local regression more accurate via a coarse-to-fine manner.

First, we discuss the case of coarse regression. For a certain local patch, we assume that the patch contains the upper limit of the crowd as  $C_m$ . Thus, the number of people in this patch is considered to be  $[0, C_m]$ . We equally divide  $[0, C_m]$  into  $s$  intervals and the length of each interval is  $\frac{C_m}{s}$ . The vector  $\mathbf{p} = [p_1, p_2, \dots, p_s]^T$  represents the probability of  $s$  intervals, and the vector  $\mathbf{v} = [v_1, v_2, \dots, v_s]^T = [\frac{1 \cdot C_m}{s}, \frac{2 \cdot C_m}{s}, \dots, C_m]^T$  represents the value of  $s$  intervals.

Then the counting number  $C_p$  of a local patch in coarse regression can be obtained as followed:

$$C_p = \mathbf{p}^T \mathbf{v} = \sum_{i=1}^s p_i \cdot v_i = \sum_{i=1}^s p_i \cdot \frac{i \cdot C_m}{s} = C_m \sum_{i=1}^s \frac{p_i \cdot i}{s}. \quad (6)$$

Next, we discuss the situation of fine mixture regression. We assume that the fine regression is consisted of  $K$  mixtures. Then, the interval number of the  $k$ -th mixture is  $s_k$ . The vector  $\mathbf{p}$  of the  $k$ -th mixture is  $\mathbf{p}_k = [p_{k,1}, p_{k,2}, \dots, p_{k,s}]^T$  and the vector  $\mathbf{v}$  is  $\mathbf{v}_k = [v_{k,1}, v_{k,2}, \dots, v_{k,s}]^T = [\frac{1 \cdot C_m}{\prod_{j=1}^k s_j}, \frac{2 \cdot C_m}{\prod_{j=1}^k s_j}, \dots, \frac{s_k \cdot C_m}{\prod_{j=1}^k s_j}]^T$ . The counting number  $C_p$  of a local patch in mixture regression can be calculated as followed:

$$C_p = \sum_{k=1}^K \mathbf{p}_k^T \mathbf{v}_k = \sum_{k=1}^K (\sum_{i=1}^{s_k} p_{k,i} \cdot \frac{i_k \cdot C_m}{\prod_{j=1}^k s_j}) = C_m \sum_{k=1}^K \sum_{i=1}^{s_k} \frac{p_{k,i} \cdot i_k}{\prod_{j=1}^k s_j}. \quad (7)$$

To illustrate the operation of MRM clearly, we take the regression with three mixtures ( $K=3$ ) for example. For the first mixture, the length of each interval is  $C_m/s_1$ . The interval is roughly divided, and the network learns a preliminary estimation of the degree of density, such as sparse, medium, or dense. As the deeper feature in the network contains richer semantic information, we adopt the feature map of layer 5 for the first mixture. For the second and third mixtures, the length of each interval is  $C_m/(s_1 \times s_2)$  and  $C_m/(s_1 \times s_2 \times s_3)$ , respectively. Based on the fine estimation of the second and third mixtures, the network performs more accurate and detailed regression. Since the shallower features in the network contain detailed texture information, we exploit the feature maps of layer 4 and layer 3 for the second and third mixtures of counting regression, respectively.

### 3.4 Adaptive Soft Interval Module

In Sect. 3.3, it is very inflexible to directly divide the regression interval into several non-overlapping intervals. The regression of value at hard-divided interval boundary will cause a significant error. Therefore, we propose ASIM, which can shift and scale interval adaptively to make the regression process smooth.

For shifting process, we add an extra interval shifting vector factor  $\beta_k = [\beta_{k,1}, \beta_{k,2}, \dots, \beta_{k,s}]^T$  to represent interval shifting of the  $i$ -th interval of the  $k$ -th mixture, and the index of the  $k$ -th mixture  $\bar{i}_k$  can be updated to  $\bar{i}_k = i_k + \beta_{k,i}$ .

For scaling process, similar to the shifting process, we add an additional interval scaling factor  $\gamma$  to represent interval scaling of each mixture, and the interval number of the  $k$ -th mixture  $\bar{s}_k$  can be updated to  $\bar{s}_k = s_k(1 + \gamma_k)$ .

The network can get the output parameters  $\{\mathbf{p}_k, \gamma_k, \beta_k\}$  for an input image. Based on Eq. (7) and the given parameters  $C_m$  and  $s_k$ , we can update the mixture regression result  $C_p$  to:



$$C_p = C_m \sum_{k=1}^K \sum_{i=1}^{s_k} \frac{p_{k,i} \cdot \bar{i}_k}{\prod_{j=1}^k \bar{s}_j} = C_m \sum_{k=1}^K \sum_{i=1}^{s_k} \frac{p_{k,i} \cdot (i_k + \beta_{k,i})}{\prod_{j=1}^k [s_j(1 + \gamma_k)]}. \quad (8)$$

Now, we detail the specific implementation of MRM and ASIM. As shown in Fig. 3, for the feature maps from SAM, we downsample them to size  $\frac{M}{w} \times \frac{N}{h}$  by following a two-stream model ( $1 \times 1$  convolution and avg pooling,  $1 \times 1$  convolution and max pooling) and channel-wise concatenation operation. In this way, we can get the fused feature map from the two-stream model to avoid excessive information loss caused via down-sampling. With linear mapping via  $1 \times 1$  convolution and different activation functions (ReLU, Tanh and Sigmoid), we get regression factors  $\{\mathbf{p}_k, \gamma_k, \beta_k\}$ , respectively. We should note that,  $\{\mathbf{p}_k, \gamma_k, \beta_k\}$  are the output of MRM and ASIM modules, only related to the input image. LCM is calculated according to parameters  $\{\mathbf{p}_k, \gamma_k, \beta_k\}$  and point-wise operation in Eq. (8). Crowd number can be calculated via global summation over the LCM. The entire network can be trained end-to-end. The target of network optimization is  $L_1$  distance between the estimated LCM ( $LCM^{es}$ ) and the ground-truth LCM ( $LCM^{gt}$ ), which is defined as  $Loss = \|LCM^{es} - LCM^{gt}\|_1$ .

## 4 Experiments

In this section, we first introduce four public challenging datasets and the essential implementation details in our experiments. After that, we compare our method with state-of-the-art methods. Finally, we conduct extensive ablation studies to prove the effectiveness of each component of our method.

### 4.1 Datasets

We evaluate our method on four publicly available crowd counting benchmark datasets: ShanghaiTech [36] Part A and Part B, UCF-QNRF [8] and UCF-CC-50 [7]. These datasets are introduced as follows.

**ShanghaiTech.** The ShanghaiTech dataset [36] is consisted of two parts: Part A and Part B, with a total of 330,165 annotated heads. Part A is collected from the Internet and represents highly congested scenes, where 300 images are used for training and 182 images for testing. Part B is collected from shopping street surveillance camera and represents relatively sparse scenes, where 400 images are used for training and 316 images for testing.

**UCF-QNRF.** The UCF-QNRF dataset [8] is a large crowd counting dataset with 1535 high resolution images and 1.25 million annotated heads, where 1201 images are used for training and 334 images for testing. It contains extremely dense scenes where the maximum crowd count of an image can reach 12865. We resize the long side of each image within 1920 pixels to reduce cache occupancy, due to the large resolution of images in the dataset.

**UCF-CC-50.** The UCF-CC-50 dataset [7] is an extremely challenging dataset, containing 50 annotated images of complicated scenes collected from the Internet. In addition to different resolutions, aspect ratios and perspective distortions, this dataset also has great variants of crowd numbers, varying from 94 to 4543.

## 4.2 Implementation Details

**Evaluation Metrics.** We adopt mean absolute error (MAE) and mean squared error (MSE) as metrics to evaluate the accuracy of crowd counting estimation, which are defined as:

$$MAE = \frac{1}{N} \sum_{i=1}^N |C_i^{es} - C_i^{gt}|, \quad MSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (C_i^{es} - C_i^{gt})^2}, \quad (9)$$

where  $N$  is the total number of testing images,  $C_i^{es}$  (*resp.*  $C_i^{gt}$ ) is the estimated (*resp.* ground-truth) count of the  $i$ -th image, which can be calculated by summing the estimated (*resp.* ground-truth) LCM of the  $i$ -th image.

**Data Augmentation.** In order to ensure our network can be sufficiently trained and keep good generalization, we randomly crop an area of  $m \times m$  pixels from the original image for training. For the ShanghaiTech Part B and UCF-QNRF datasets,  $m$  is set to 512. For the ShanghaiTech Part A and UCF-CC-50 datasets,  $m$  is set to 384. Random mirroring is also performed during training. In testing, we use the original image to infer without crop and resize operations. For the fair comparison with the previous typical work CSRNet [13] and SANet [3], we does not add the scale augmentation during the training and test stages.

**Training Details.** Our method is implemented with PyTorch. All experiments are carried out on a server with an Intel Xeon 16-core CPU (3.5 GHz), 64 GB RAM and a single Titan Xp GPU. The backbone of network is directly adopted from convolutional layers of VGG16 [24] pretrained on ImageNet, and the other convolutional layers employ random Gaussian initialization with a standard deviation of 0.01. The learning rate is initially set to  $1e^{-5}$ . The training epoch is set to 400 and the batch size is set to 1. We train our networks with Adam optimization [10] by minimizing the loss function.

## 4.3 Comparisons with State of the Art

The proposed method exhibits outstanding performance on all the benchmarks. The quantitative comparisons with state-of-the-art methods on four datasets are presented in Table 1 and Table 2. In addition, we also tell the visual comparisons in Fig. 6.

**ShanghaiTech.** We compare the proposed method with multiple classic methods on ShanghaiTech Part A & Part B dataset and it has significant performance improvement. On Part A, our method improves 9.69% in MAE and 14.47% in

**Table 1.** Comparisons with state-of-the-art methods on ShanghaiTech Part A and Part B datasets

Dataset	Part A		Part B	
	MAE	MSE	MAE	MSE
MCNN [36]	110.2	173.2	26.4	41.3
Switch-CNN [2]	90.4	135.0	21.6	33.4
CP-CNN [25]	73.6	106.4	20.1	30.1
CSRNet [13]	68.2	115.0	10.6	16.0
SANet [3]	67.0	104.5	8.4	13.6
PACNN [22]	66.3	106.4	8.9	13.5
SFCN [31]	64.8	107.5	7.6	13.0
Encoder-Decoder [9]	64.2	109.1	8.2	12.8
CFF [23]	65.2	109.4	7.2	12.2
Bayesian Loss [17]	62.8	101.8	7.7	12.7
SPANet+CSRNet [6]	62.4	99.5	8.4	13.2
RANet [34]	59.4	102.0	7.9	12.9
PaDNet [29]	<b>59.2</b>	<b>98.1</b>	8.1	12.2
Ours	61.59	98.36	<b>7.02</b>	<b>11.00</b>

MSE compared with CSRNet, improves 8.07% in MAE and 5.42% in MSE compared with SANet. On Part B, our method improves 33.77% in MAE and 31.25% in MSE compared with CSRNet, improves 16.43% in MAE and 19.12% in MSE compared with SANet.

**UCF-QNRF.** We then compare the proposed method with other related methods on the UCF-QNRF dataset. To the best of our knowledge, UCF-QNRF is currently the largest and most widely distributed crowd counting dataset. Bayesian Loss [17] achieves 88.7 in MAE and 154.8 in MSE, which currently maintains the highest accuracy on this dataset, while our method improves 2.37% in MAE and 1.68% in MSE, respectively.

**UCF-CC-50.** We also conduct experiments on the UCF-CC-50 dataset. The crowd numbers in images vary from 96 to 4633, bringing a great challenging for crowd counting. We follow the 5-fold cross validation as [7] to evaluate our method. With a small amount of training images, our network can still converge well in this dataset. Compared with the latest method Bayesian Loss [17], our method improves 19.76% in MAE and 18.18% in MSE and achieves the state-of-the-art performance.

#### 4.4 Ablation Studies

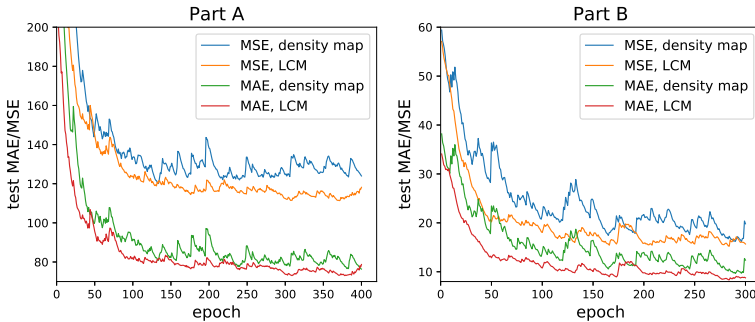
In this section, we perform ablation studies on ShanghaiTech dataset and demonstrate the roles of several modules in our approach.

**Table 2.** Comparisons with state-of-the-art methods on UCF-QNRF and UCF-CC-50 datasets

Dataset	UCF-QNRF		UCF-CC-50	
Method	MAE	MSE	MAE	MSE
MCNN [36]	277	426	377.6	509.1
Switch-CNN [2]	228	445	318.1	439.2
Composition Loss [8]	132	191	–	–
Encoder-Decoder [9]	113	188	249.4	354.5
RANet [34]	111	190	239.8	319.4
S-DCNet [32]	104.4	176.1	204.2	301.3
SFCN [31]	102.0	171.4	214.2	318.2
DSSINet [14]	99.1	159.2	216.9	302.4
MBTTBF [27]	97.5	165.2	233.1	300.9
PaDNet [29]	96.5	170.2	185.8	278.3
Bayesian Loss [17]	88.7	154.8	229.3	308.2
Ours	<b>86.6</b>	<b>152.2</b>	<b>184.0</b>	<b>265.8</b>

**Table 3.** An quantitative comparison with two different targets on testing datasets between LCM and density map

Target	Part A		Part B	
	MAE	MSE	MAE	MSE
Density map	72.98	114.89	9.79	14.40
Local counting map	<b>69.52</b>	<b>110.23</b>	<b>8.96</b>	<b>13.51</b>

**Fig. 4.** The curves of testing loss for different regression targets LCM and density map. LCM has lower error and smoother convergence curves on both MAE and MSE than density map

**Effect of Regression Target.** We analyze the effects of different regression targets firstly. As shown in Table 3, the LCM we introduced has better

**Table 4.** Ablation study on different combinations of modules including MRM, ASIM and SAM in the regression framework

Module	Part A		Part B	
	MAE	MSE	MAE	MSE
LCM	69.52	110.23	8.96	13.51
MRM	65.24	104.81	7.79	12.55
MRM+ASIM	63.85	102.48	7.56	11.98
MRM+ASIM+SAM	<b>61.59</b>	<b>98.36</b>	<b>7.02</b>	<b>11.00</b>

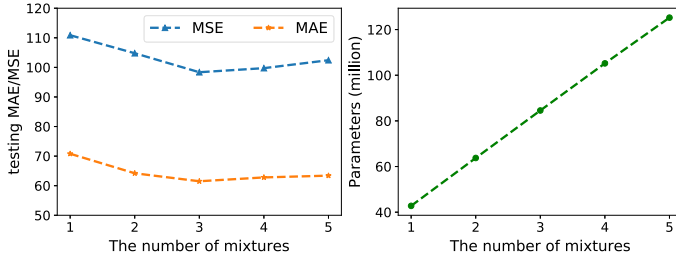
performance than the density map, with 4.74% boost in MAE and 4.06% boost in MSE on Part A, 8.47% boost in MAE and 6.18% boost in MSE on Part B. As shown in Fig. 4, LCM has more stable and lower MAE & MSE testing curves. It indicates that LCM alleviates the inconsistency problem between the training target and the evaluation criteria to bring performance improvement. Both of them adopt VGG16 as the backbone networks without other modules.

**Effect of Each Module.** To validate the effectiveness of several modules, we train our model with four different combinations: 1) VGG16+LCM (Baseline); 2) MRM; 3) MRM+ASIM; 4) MRM+ASIM+SAM. As shown in Table 4, MRM improves the MAE from 69.52 to 65.24 on Part A and from 8.96 to 7.79 on Part B, compared with our baseline direct LCM regression. With ASIM, it improves the MAE from 65.24 to 63.85 on Part A and from 7.79 to 7.56 on Part B. With SAM, it improves the MAE from 63.85 to 61.59 on Part A and from 7.56 to 7.02 on Part B, respectively. The combination of MRM+ASIM+SAM achieves the best performance, 61.59 in MAE and 98.36 in MSE on Part A, 7.02 in MAE and 11.00 in MSE on Part B.

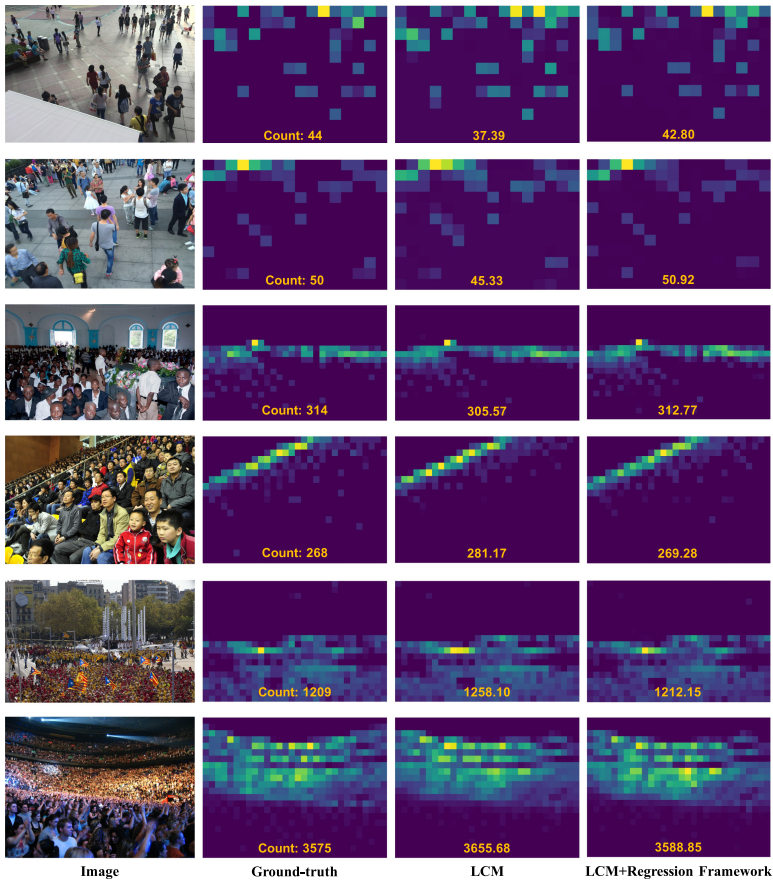
**Table 5.** The effects of different local patch sizes with MRM module

Size	Part A		Part B	
	MAE	MSE	MAE	MSE
$16 \times 16$	70.45	114.12	9.41	13.93
$32 \times 32$	69.28	109.24	8.68	13.44
$64 \times 64$	<b>65.24</b>	<b>104.81</b>	<b>7.79</b>	<b>12.55</b>
$128 \times 128$	67.73	105.15	7.93	12.78

**Effect of Local Patch Size.** We analyze the effects of different local patch sizes on regression results with MRM. As shown in Table 5, the performance gradually improves with local patch size increasing and it slightly drops until  $128 \times 128$  patch size. Our method gets the best performance with  $64 \times 64$  patch size on Part A and Part B. When the local patch size is too small, the heads information that local patch can represent is too limited, and it is difficult to



**Fig. 5.** The left figure shows the MAE & MSE performance of different mixtures number on ShanghaiTech Part A. The right one shows the relationship between the number of mixtures and the parameters of mixtures



**Fig. 6.** From top to bottom, we exhibit several example images with different densities from sparse, medium to dense. The second, third and fourth columns display ground-truth LCM, LCM generated with the baseline model (VGG16+LCM) and LCM generated with our proposed regression framework, respectively.

map weak features to the counting value. When the local patch size is  $1 \times 1$ , the regression target changes from LCM to the density map. When the local patch size is too large, the counting regression range will also expand, making it difficult to perform fine and accurate estimation.

**Effect of Mixtures Number  $K$ .** We measure the performance of adaptive mixture regression network with different mixture numbers  $K$ . As shown in Fig. 5, the testing error firstly drops and then slightly arises with the increasing number of  $K$ . On the one hand, smaller  $K$  (e.g.,  $K = 1$ ) means single division and it will involve a coarse regression on the local patch. On the other hand, larger  $K$  (e.g.,  $K = 5$ ) means multiple divisions. It's obviously unreasonable when we divide each interval via undersize steps, such as 0.1 and 0.01. The relationship between the number of mixtures and model parameters is shown in Fig. 5. To achieve a proper balance between the accuracy and computational complexity, we take  $K = 3$  as the mixtures number in experiments.

## 5 Conclusion

In this paper, we introduce a new learning target named local counting map, and show its feasibility and advantages in local counting regression. Meanwhile, we propose an adaptive mixture regression framework in a coarse-to-fine manner. It reports marked improvements in counting accuracy and the stability of the training phase, and achieves the start-of-the-art performances on several authoritative datasets. In the future, we will explore better ways of extracting context and multi-scale information from different convolutional layers. Additionally, we will explore other forms of local area supervised learning approaches to further improve crowd counting performance.

## References

1. Babu Sam, D., Sajjan, N.N., Venkatesh Babu, R., Srinivasan, M.: 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 (2018)
2. Babu Sam, D., Surya, S., Venkatesh Babu, R.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
3. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale aggregation network for accurate and efficient crowd counting. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 757–773. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01228-1\\_45](https://doi.org/10.1007/978-3-030-01228-1_45)
4. Chattopadhyay, P., Vedantam, R., Selvaraju, R.R., Batra, D., Parikh, D.: Counting everyday objects in everyday scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
5. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: Proceedings of the British Machine Vision Conference (2012)

6. Cheng, Z.Q., Li, J.X., Dai, Q., Wu, X., Hauptmann, A.G.: Learning spatial awareness to improve crowd counting. In: Proceedings of the International Conference on Computer Vision (2019)
7. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2013)
8. Idrees, H., et al.: Composition loss for counting, density map estimation and localization in dense crowds. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 544–559. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01216-8\\_33](https://doi.org/10.1007/978-3-030-01216-8_33)
9. Jiang, X., et al.: Crowd counting and density estimation by trellis encoder-decoder networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
11. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Proceedings of the Conference and Workshop on Neural Information Processing Systems (2010)
12. Li, J., Liang, X., Shen, S., Xu, T., Feng, J., Yan, S.: Scale-aware fast R-CNN for pedestrian detection. *IEEE Tran. Multimedia* **20**(4), 985–996 (2017)
13. Li, Y., Zhang, X., Chen, D.: CSRNet: dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
14. Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W., Lin, L.: Crowd counting with deep structured scale integration network. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
15. Liu, W., Liao, S., Ren, W., Hu, W., Yu, Y.: High-level semantic feature detection: a new perspective for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
16. Lu, H., Cao, Z., Xiao, Y., Zhuang, B., Shen, C.: TasselNet: counting maize tassels in the wild via local counts regression network. *Plant Methods* **13**(1), 79 (2017)
17. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: Proceedings of the International Conference on Computer Vision (2019)
18. Mao, J., Xiao, T., Jiang, Y., Cao, Z.: What can help pedestrian detection? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
19. Paul Cohen, J., Boucher, G., Glastonbury, C.A., Lo, H.Z., Bengio, Y.: Countception: counting by fully convolutional redundant counting. In: Proceedings of the International Conference on Computer Vision (2017)
20. Sam, D.B., Babu, R.V.: Top-down feedback for crowd counting convolutional neural network. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)
21. Shang, C., Ai, H., Bai, B.: End-to-end crowd counting via joint learning local and global count. In: Proceedings of the International Conference on Image Processing (2016)
22. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
23. Shi, Z., Mettes, P., Snoek, C.G.M.: Counting with focus for free. In: Proceedings of the International Conference on Computer Vision (2019)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)



25. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid CNNs. In: Proceedings of the International Conference on Computer Vision (2017)
26. Sindagi, V.A., Patel, V.M.: HA-CNN: hierarchical attention-based crowd counting network. *IEEE Trans. Image Process.* **29**, 323–335 (2019)
27. Sindagi, V.A., Patel, V.M.: Multi-level bottom-top and top-bottom feature fusion for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
28. Stahl, T., Pinteá, S.L., van Gemert, J.C.: Divide and count: generic object counting by image divisions. *IEEE Trans. Image Process.* **28**(2), 1035–1044 (2018)
29. Tian, Y., Lei, Y., Zhang, J., Wang, J.Z.: PaDNet: pan-density crowd counting. *IEEE Trans. Image Process.* **29**, 2714–2727 (2020)
30. Wan, J., Chan, A.: Adaptive density map generation for crowd counting. In: Proceedings of the International Conference on Computer Vision (2019)
31. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
32. Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., Shen, C.: From open set to closed set: counting objects by spatial divide-and-conquer. In: Proceedings of the International Conference on Computer Vision (2019)
33. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to scale: generating multi-polar normalized density maps for crowd counting. In: Proceedings of the International Conference on Computer Vision (2019)
34. Zhang, A., et al.: Relational attention network for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
35. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
36. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)