Crowd Counting via Multi-layer Regression

Xin Tan[1], Chun Tao[2], Tongwei Ren[1,*], Jinhui Tang[3], and Gangshan Wu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Nanjing Tech University, China
[3]Nanjing University of Science and Technology, China
tx@smail.nju.edu.cn, taochunwx@foxmail.com, rentw@nju.edu.cn
jinhuitang@njust.edu.cn, gswu@nju.edu.cn

ABSTRACT

Crowd counting aims to estimate the number of persons in a crowd image–a challenge until this day–as congestion degree varies, people's appearances may seem different. To address this problem, we propose a novel crowd counting method named Multi-layer Regression Network (MRNet), which consists of a multi-layer recognition branch and several density regressors. In practice, the recognition branch recognizes the congestion degree of the regions in a crowd image, then disintegrates the image into background and several crowd regions layer by layer, each regions are assigned different congestion degrees. In each layer, the recognized crowd regions with the specific congestion degree are delivered to a regressor with the corresponding density prior for crowd density estimation. The generated density maps at all layers are integrated to obtain the final density map for crowd density estimation. To date, MRNet is the first method to estimate crowd densities on crowd regions with different regressors. We conduct a comprehensive evaluation of MRNet on four typical datasets in comparison with nine state-of-the-art methods. By using multi-layer regression, MRNet achieves significant improvement in crowd counting accuracy, and outperforms the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies → Computer vision**.

KEYWORDS

Crowd counting; multi-layer regression; recognition branch; density regressor; congestion degree

ACM Reference Format:
Xin Tan, Chun Tao, Tongwei Ren, Jinhui Tang, and Gangshan Wu. 2019. Crowd Counting via Multi-layer Regression. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), Oct. 21–25, 2019, Nice, France.* ACM, NY, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350914

MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350914

1 INTRODUCTION

Crowd counting aims to count the number of persons in a crowd image, which has drawn much attention because of its wide application in public security, crowd monitoring and behavior analysis [4]. As shown in Figure 1, as the crowd becomes congested, people's appearances may become harder to recognize, therefore causing difficulty for face or person detection method. [6, 10, 30]. Recently, the typical solution for crowd counting is to estimate crowd density with Convolutional Neural Networks (CNNs) based density regression [3, 15, 34], in which CNNs are trained to learn the mapping between crowd images and density maps, and the number of persons are acquired via the sum of density maps. Figure 1 illustrates some examples of density map, where the sparse regions are labelled in blue and congested regions in red.

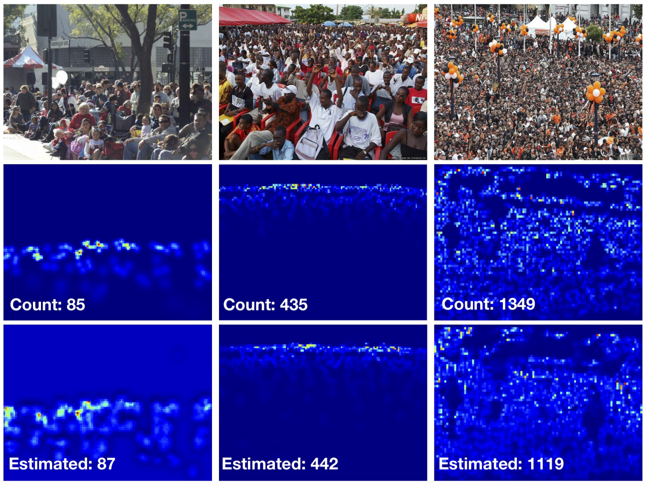

Figure 1: Examples of crowd counting with crowd density estimation. From top to bottom: crowd images, ground truths, and density maps generated by MRNet.

Recent years have witnessed significant advances in person counting and crowd density estimation, especially in solving scale variation issue by extracting different features with multi-scale architectures [2, 19, 23, 27, 34]. For example, MCNN uses a three-column network, in which each column is dedicated to a certain layer of a congested scene, to extract features in different scales [34]. Meanwhile, deep network architectures are developed for crowd counting, *e.g.*, CSRNet [15] demonstrates its advantages in obtaining more concise and effective results than multi-column architectures.

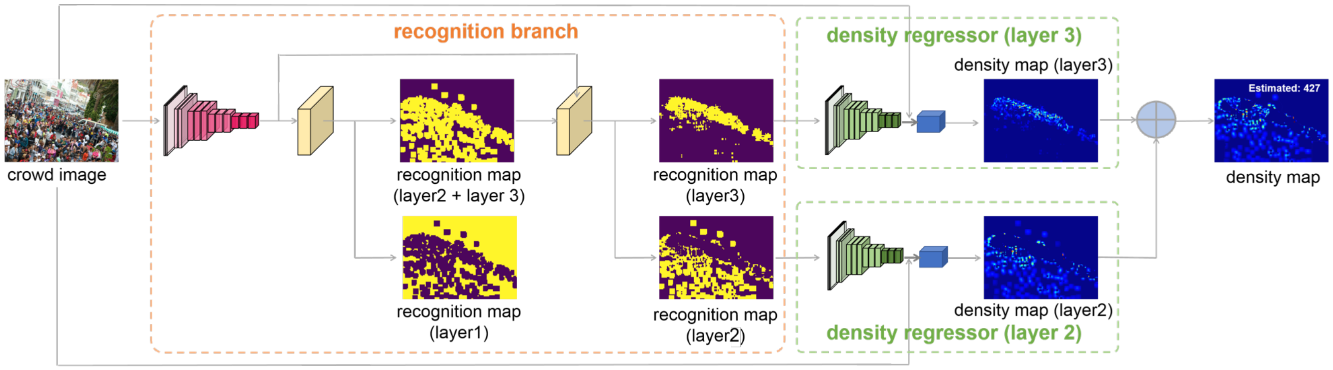

Figure 2: An overview of the proposed MRNet.

However, it remains a thorny problem to provide accurate crowd density estimation, as the crowd appearances vary as a result of different perspective distortion, occlusion and scale variation (as shown in Figure 1). Specifically, crowd appearance in congested scenes is quite different to that in sparse scenes, because most persons in congested scenes may not be completely visible. Normally, with the increase of congestion degree, the scale of person gets smaller, the density per unit area increases and the person's appearance becomes incomplete. Such differences in crowd appearances cannot be effectively handled by extracting features from different scales. It is required to extract different features in regions with diverse congestion degrees to represent persons, which is not a major concern in previous researches on crowd counting. What ADCrowdNet [17] proposed bears a close resemblance, which assigns different weights to represent the person in the image; it also attempts to binarize the weights by applying a threshold and to divide a crowd image into regions with people and regions without, and only estimates crowd density in the former.

To address the problem of congestion degree diversity, we propose a novel crowd counting method named *Multi-layer Regression Network* (MRNet), which consists of a multi-layer recognition branch and several density regressors. The working principles of MRNet is to improve the accuracy of crowd counting by applying multiple regressors on regions with different congestion degrees. To date, MRNet is the first of its kind to generate density maps with multiple regression functions.

Figure 2 shows the 3-layer architecture of MRNet. For a given crowd image, we propose a recognition branch to segment it into the regions with different congestion degrees gradually by learning features representing congestion degree. At each layer, the recognized regions are delivered to a density regressor with the corresponding density prior, and the remaining parts of the crowd image are delivered to the next layers with other regressors. Because each regressor focuses on crowd density estimation with the similar congestion degrees, it is able to provide more accurate estimation by optimizing specific parameters of its mapping function as compared to previous methods of a general regressor for the entire image. Finally, the density maps generated by all the regressors are integrated to count the number of persons in the crowd image. By leveraging different features on the regions with different congestion degrees, our method is more adaptable to a variety of crowd images. The experimental results show that our method outperforms the state-of-the-art methods on four typical datasets.

The main contributions of this paper include:

- The proposal of the first crowd counting method to solve the problem of congestion degree diversity, which is an essential element of the accuracy of predicting density maps.
- The proposal of a novel multi-layer regression network–consisting of a multi-layer recognition branch and multiple density regressors-to generate density maps for regions with different congestion degrees separately.
- Evaluation of MRNet's performance on four typical datasets, and its superiority over state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, we survey the existing crowd counting methods based on detection, regression and CNNs. In Section 3, we present the details of MRNet, including recognition branch, density regressor, and the procedures of ground truth generation and training. In Section 4, we evaluate MRNet on four typical datasets, and compare it with the state-of-the-art methods. Section 5 comprises of conclusion of the paper.

2 RELATED WORK

Previous algorithms for crowd counting could be divided into three categories: detection-based methods, regression-based methods and CNN-based methods.

2.1 Detection-based Methods

Earlier approaches, to a large extent focus on detection [9] and counting with low-level features extracted from people. A common way of detection is to use a sliding window applied with a well-trained classifier to detect the whole human body [6, 30] or body part [10], and it achieves reasonable accuracy when crowd is sparse. However, congestion in crowd and obscurity with occlusion and scene clutter compromise accuracy of counting because human facial or body features are not complete or hard to extract.

2.2 Regression-based Methods

Due to the limitation of applying detection-based method in congested scenes, regression-based methods [7, 11] are developed to avoid complicated object detection problems. Regression-based methods try to directly learn the mapping between extracted features of the images and actual object counts, showing little consideration for object location information. But this comes at the cost of increased annotation effort to label a huge amount of dotted annotations or positions in the training images. Lempitsky *et al.* [14] propose a method to learn a linear mapping between features of the local regions and density at each pixel, which can perform counting at arbitrary locations. The integral of all the pixels in the image denotes the count of objects. Considering the difficulty in obtaining an ideal linear mapping, Pham *et al.* [20] propose a random forest to learn a non-linear mapping between features and image.

2.3 CNN-based Methods

Researchers also focus on CNN-based methods to estimate crowd density map because of its advantage record in visual recognition and classification. Walach *et al.* [31] apply layered boosting and selective sampling to their work. Shang *et al.* [24] use end-to-end CNN to estimate the local and global counts concurrently, which directly generates the final people counts by inputting the entire original image. Boominathan *et al.* [2] use deep and shallow networks to capture both the high-level semantic information and the low-level features for generating density map. Zhang *et al.* [34] design MCNN, CNN with multi-column architecture, to tackle the large scale variation in crowd scenes by extracting features at different scales. Similar to MCNN, Onoro and Sastre [19] present a scale-aware network with multiple columns, called Hydra, to extract features at different scales. Sindagi *et al.* [26] incorporate a high-level prior into the density estimation network to boost the prediction performance. Li *et al.* [15] use deep convolutional networks with dilated convolution layers [6] to enlarge the receptive field and extract deeper information. To capture inevitable density variation, DecideNet [16] estimates the crowd density by generating object detection [22] and regression based density maps separately. Cao *et al.* [3] use scale aggregation modules, which follow the idea of [28, 29], to extract multi-scale features and transposed convolutions to generate high resolution density map. Liu *et al.* [17] propose an attention-aware network, leveraging semantic segmentation [1, 18, 21, 33] to provide crowd location information to density map estimator, which not only obtains congestion degree value but also blocks out the background noise.

3 MULTI-LAYER REGRESSION NETWORK

Figure 2 demonstrates the network structure of our proposed method. The MRNet comprises two components: recognition branch and density regressors. The recognition branch is based on fully convolutional network and used to detect crowd regions with different congestion degrees layer by layer, while multiple density regressors employ deep CNN to generate density maps for each crowd region. In order to avoid the confusion caused by predicting different kinds of congested crowd at the same time, classification strategy between binary and multi-class classification must be taken into account as the crowd may be omitted or wrongly classified into the category with similar congestion degree. After a series of attempts, it comes out with the conclusion that the multi-layer binary classification strategy outperforms the multi-class classification. In the regression stage, density maps independently generated from multiple density regressors are used in final result fusion. Every density regressor consists of a VGG-16 [25] backbone network pretrained on ImageNet [8] to extract the image features, and a backend network which encompasses pure convolutional layer to map the feature into the density map. After feature extraction, feature maps generated by backbone multiply the corresponding classification result pixel-wisely to filter out the noises and irrelevant crowd regions. The filtered maps which consist of crowd regions are delivered to the backend network for next regression step. Finally, all predicted density maps are summed to obtain the final result.

3.1 Recognition Branch

The purpose of the recognition branch is to discover the location and congestion degree of the gathering. The boundaries between crowd density levels are not distinct, making it hard for the model to segment the crowd in regions with varying congestion degrees as the crowd may be classified to the wrong category because its congestion degree is on the verge of the boundary. To tackle difficulties and eliminate uncertainties, recognition branch applies multi-layer disintegration strategy to segment the crowd in lieu of multi-class classification. Each layer of the recognition branch recognizes crowd regions with specific congestion degree and delivers the unidentified region to the next layer for further recognition. For instance, the first layer of recognition branch performs binary classification on the original image in which one of the classes stands for background and the other represents the area where the crowd density is greater than 0. The next recognition layer of the branch performs binary classification on the detected crowd regions which are to be further identified. New classification result includes the regions that should be delivered to the next layer and those belong to this layer. Once the congestion degree of the region is determined, it is separated from the image and the remaining are sent to the next recognition layer until all the crowd regions are recognized and assigned a congestion degree. Finally, recognition results are delivered to density regressors to generate density maps. Through repetition, multi-class classification can be disintegrated into multiple binary classifications in a way to avoid accuracy degradation caused by imbalanced training samples.

Our Recognition branch has two parts: frontend and backend. The frontend uses first 10 convolutional layers of VGG-16 [25] to extract crowd features, while the recognition backend, which consists of several residual blocks, recognizes and segments the crowd. The architecture of recognition branch and the detailed parameters of the backend of the recognition branch are shown in Figure 3. The output channels of each residual block are 256, 128 and 64, followed by a 1×1 convolutional layer as output layer.

3.2 Density Regressor

Inspired by the idea of data driven, we apply several independent density regressors to different crowd regions instead of using single regressor for the entire image. The number of density regressors corresponds with that of layers of recognition branch. Independent density regressor shows its advantage in avoiding the influence caused by crowd densities variation and different characteristics

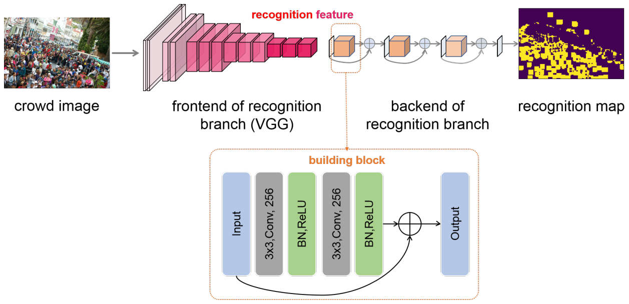

Figure 3: The architecture of the recognition branch.

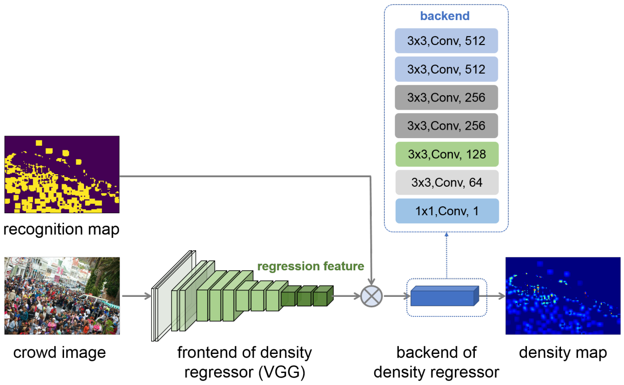

Figure 4: The architecture of a density regressor.

of noises. To a certain extent, density regressor shares a similar structure with the layer of recognition branch, both encompassing the frontend-backend pattern and a fine-tuned VGG-16 [25] as the frontend to extract low-level features. The architecture of regressor are shown in Figures 4. The recognition result is used to filter the output of the backbone which contains the image features by pixel-wise product operation, letting regressors learn the features of specific crowd regions.

In Figure 5, density regressors trained at different recognition layers generate varying regression response maps, each matching the crowd regions with different congestion degree in the given image. The regressor in the congested regions is forced to reach a relatively higher response than that in sparse regions. The accuracy of estimation on each part of the image is improved by decoupling density regression via multiple regressors.

3.3 Ground truth Generation

3.3.1 Density map. Following the method described in [34], we use the geometry-adaptive kernels to generate density map for congested scenes. The geometry-adaptive kernel is defined as:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) G_{\sigma_i}(x), \text{with } \sigma_i = \beta \overline{d_i}. \qquad (1)$$

For each annotated person x_i, the average distance of k nearest neighbors is shown as $\overline{d_i}$. Then the delta function $\delta(x - x_i)$ is convolved with a Gaussian kernel with σ_i standard deviation to generate density maps. We follow the procedure in [15] to generate the ground truth of density map, as shown in Table 1.

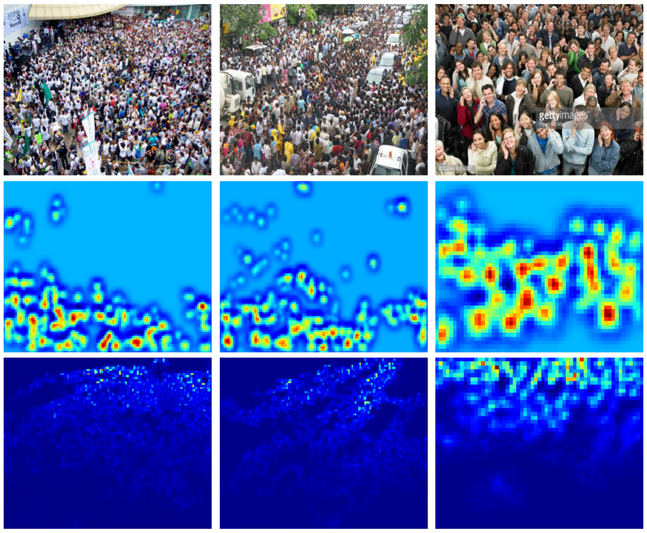

Figure 5: Examples of density maps generated by different density regressors. From top to bottom: crowd images, and density maps generated by the regressors on layer 2 and layer 3 in MRNet, respectively.

Table 1: Setups to generate density maps for different datasets.

Dataset	Generation
ShanghaiTech Part_A [34]	Geometry-adaptive kernels
UCF_CC_50 [12]	
ShanghaiTech Part_B [34]	Fixed kernel: $\sigma = 15$
UCF-QNRF [13]	
WorldExpo'10 [5]	Fixed kernel: $\sigma = 3$

3.3.2 Recognition map. It is time-consuming to use k nearest neighbors because the value of every pixel for recognition maps needs to be computed. Here we use sliding window scheme to generate recognition map, which is very easy to implement with a convolutional filter. First, the value of each pixel is set to the number of annotations of people's heads appears in the window. Then we set a threshold to convert the head counts to degrees of congestion. If the values exceed the threshold, they are set to a fixed degree of congestion, and vice versa. In a 2-layer recognition branch, we set 3 as the threshold, which only classifies sparse and congested crowds. Different configurations used to generate recognition map for different datasets are shown in Table 2.

3.4 Training

We train our method in an end-to-end way. The VGG-16 frontend is fine-tuned from a pre-trained VGG weight. Stochastic gradient descent is used as optimization method to train our model with a learning rate of 1e-6 for the density regressor and a learning rate of 5e-3 for the recognition branch. For recognition branch, we apply

Table 2: Setups to generate recognition maps for different datasets.

Dataset	Window size	Threshold
ShanghaiTech Part_A	48	3
ShanghaiTech Part_B	48	3
UCF_CC_50	80	3
UCF-QNRF	48	3
WorldExpo'10	24	3

cross-entropy loss as the loss function to evaluate the performance of crowd recognition. For density regression, we use Euclidean distance to measure the difference between the output density map and the ground truth. The loss function is defined as follows:

$$\mathcal{L}_E(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} ||F(X_i;\Theta) - F_i^g||_2^2, \tag{2}$$

where N is the batch size; $F(X_i;\Theta)$ is the predicted density map generated by density regressor with parameters denoted as Θ; X_i represents the input image; F_i^g is the ground truth of the input image.

By adding the above loss functions, our final objective function is defined as follows:

$$\mathcal{L} = \mathcal{L}_E + \sum_{i=1}^{N} \mathcal{L}_C, \tag{3}$$

where N is the layer number, and L_C denotes the cross-entropy loss mentioned above.

4 EXPERIMENTS

4.1 Datasets and Experiment Settings

We evaluate our method on four mainstream datasets: ShanghaiTech [34], UCF_CC_50 [12], UCF-QNRF [13], and WorldExpo'10 [5].

ShanghaiTech dataset. ShanghaiTech dataset is a large-scale crowd counting dataset, which contains 1,198 images with 330,165 annotated persons. The dataset is divided into two subsets: SHT Part_A and SHT Part_B. Specifically, the images in SHT Part_A are collected from websites, among which 300 images are used for training and 182 images for testing. The images in SHT Part_B have relatively sparse crowd scene and are collected from streets. 400 of them are used for training and the remaining for testing.

UCF_CC_50 dataset. UCF_CC_50 dataset includes 50 images from highly congested scenes. The annotated number of persons per image ranges from 94 to 4,543. Due to limitations in the number of images and the large span of people count, it is difficult, if not impossible, to run an accurate count. In our experiments, we adopt the standard settings [12] to run a 5-fold cross validation.

UCF-QNRF dataset. UCF-QNRF dataset is the largest real world dataset, which contains 1,535 images collected from the internet with 1,251,642 annotated persons. The annotated number of persons per image ranges from 49 to 12,865 with an average of 815 persons. The images are divided into the training dataset with 1,201 images and the testing dataset with 334 images. The average resolution of the images in UCF-QNRF dataset is 2013×2902. To reduce training time and pressure on memory capacity, we reduce the length of the longer side of the images to 1024 and scale down the other proportionally.

WorldExpo'10 dataset. WorldExpo'10 dataset contains 3,980 frames from 1,132 video sequences captured by 108 surveillance cameras, among which 3,380 frames are used for training and the remaining from 5 different scenes–120 frames per scene–are used for testing. It also provides the regions of interest and the perspective map for each frame, and we use the regions of interest in our experiments.

All the experiments are conducted on the same computer with i7 3.5GHz CPU, 32GB memory and 1080Ti GPU.

4.2 Evaluation Metrics

Mean absolute error (MAE) and mean square error (MSE) are two widely used metrics in crowd counting evaluation. Their definitions are shown as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |C_i - C_i^g|, \tag{4}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (C_i - C_i^g)^2}, \tag{5}$$

where N is the number of test images; C_i denotes the estimated number of persons in the ith image, and C_i^g denotes the ground truth of the ith image, *i.e.*, the annotated number of persons in the ith image.

Table 3: Evaluation of MRNet with different multi-layer disintegration strategies on SHT Part_A dataset.

Configuration	MAE	MSE
MRNet (2-layer)	65.8	106.5
MRNet (3-layer)	**63.3**	**97.8**
MRNet (4-layer)	66.4	108.0
MRNet (3-class)	67.1	108.8

4.3 Component Analysis

Recognition branch is a key module in MRNet, which disintegrates a crowd image into regions with different congestion degrees and feeds them into corresponding density regressors. There are two important components in the construction of recognition branch. One of them is the strategy to disintegrate a crowd image into multiple regions, *i.e.*, the number of used layers and how to classify those regions. The other is the generation of recognition map, especially the window size used in label generation.

4.3.1 Multi-layer disintegration strategy. We study the influence of the number of layers in crowd image disintegration on the accuracy of crowd counting on SHT Part_A dataset. We also compare the effectiveness of multi-layer binary classification and one-layer multi-class classification in crowd image disintegration.

The top three rows in Table 3 show the accuracy of crowd counting via multi-layer binary classification with different numbers of layers. Specifically, the 2-layer MRNet disintegrates the crowd

Table 4: Evaluation of MRNet trained with the recognition maps generated with different window sizes on UCF_CC_50 dataset.

Window Size	MAE	MSE
32	261.4	375.0
64	252.7	342.1
80	**232.3**	**314.8**
128	247.3	344.3

image into regions of crowd and background without persons, and the regions of crowd are estimated with the same regressor; the 3-layer MRNet further disintegrates the regions of crowd into two categories, sparse and congested, and feeds the regions to its corresponding density regressor; the 4-layer MRNet disintegrates the regions of crowd into three categories–low-congested, mid-congested and high-congested–as compared to the two categories in the 2-layer MRNet. We train the methods with different layers following the procedure described in Section 3.4. To disintegrate the regions of crowd into sparse regions and congested regions in 3-layer MRNet, window size is set at 72 and threshold at 3. Sparse regions in 3-layer MRNet is compatible to low-congested regions in 4-layer MRNet. Congested regions, on the other hand, is further distinguished between mid-congested regions and high-congested regions at the same window size of 72 and threshold of 6. As shown in Table 3, 2 conclusions may be drawn therefrom: 1) The fact that 3-layer MRNet obtains better performance than the 2-layer MRNet illustrates the effectiveness of multi-layer regression rather than simply disintegrating crowd images into crowd and background as in ADCrowdNet [17]; 2) The result that 3-layer MRNet obtains better performance than 4-layer MRNet may point to the following suggestion: the increase of the number of layers may not be positively correlated to the accuracy of crowd counting. A possible explanation is that as a result of unclear boundary in between, the imprecise division in the category of congestion degree may occur.

We also validate the performance of directly disintegrating crowd images into three categories, which is denoted as MRNet (3-class) in Table 3. We can see that the performance of the 3-class MRNet is 6.0% higher than in MAE and 4.6% higher in MSE, as compared to that of the 3-layer MRNet. It may be caused by the imbalance in training samples of different categories of regions, *i.e.*, the high-congested regions are much less than those with low crowd densities (as shown in Figure 6), which is another contributor to the inaccuracy in mid-congested and high-congested region disintegration in the 4-layer MRNet.

4.3.2 Window size in ground truth generation. We study the influence of different window sizes in generating the ground truths to the accuracy of crowd counting on UCF_CC_50 dataset. According to Table 2, we use 80 as the default window size and select three widely used window sizes, 32, 64, and 128, for comparison. Figure 7 shows the examples of the recognition maps generated with these different window sizes. In the generation of all the recognition maps, the threshold is set to 3.

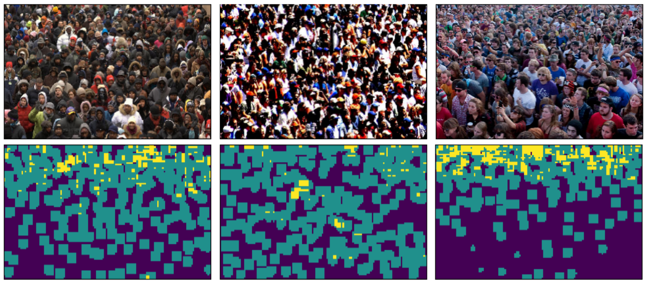

Figure 6: Examples of imbalanced distribution of recognition maps on SHT Part_A dataset. From top to bottom: crowd images, and recognition maps in ground truth. Here, the background, the sparse regions and the congested regions are labelled in black, yellow and green, respectively.

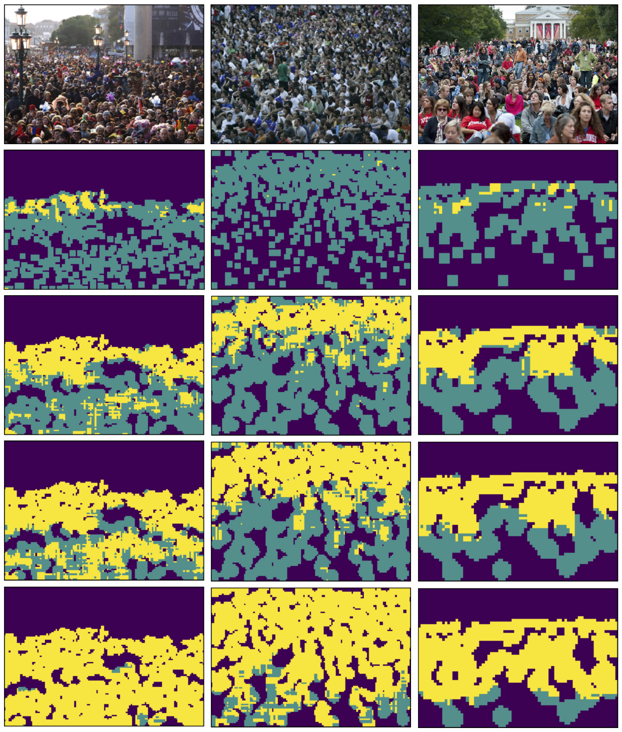

Figure 7: Examples of the recognition maps generated with different window sizes. From top to bottom: crowd images from SHT Part_A dataset, and recognition maps in ground truth generated with the window sizes of 32, 64, 80 and 128, respectively.

Table 4 shows the performance of crowd counting when MRNet is trained with the recognition maps generated with different window sizes. We can see that MRNet with the default window size obtains the best performance. The experimental results show that a fast and feasible way to determine the window size is to use the average value of one-tenth of the shorter side of all the images in a dataset. As shown in Figure 7, at a smaller window size (*i.e.*,

Table 5: Evaluation of different methods on ShanghaiTech, UCF_CC_50, UCF-QNRF and WorldExpo'10 datasets. The results of the first place on each metric are marked in red and second place in blue.

Method	SHT Part_A		SHT Part_B		UCF_CC_50		UCF-QNRF		WorldExpo'10	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [34]	110.2	173.2	26.4	41.3	377.6	509.1	277.0	426.0	11.6	-
Cascaded-MTL [26]	101.3	152.4	20.0	31.1	322.8	397.9	252.0	514.0	-	-
Switching-CNN [23]	90.4	135.0	21.6	33.4	318.1	439.2	228.0	445.0	9.4	-
CP-CNN [27]	73.6	106.4	20.1	30.1	295.8	320.9	-	-	8.9	-
CSRNet [15]	68.2	115.0	10.6	16.0	266.1	397.5	-	-	8.6	-
CL [13]	-	-	-	-	-	-	132.0	191.0	-	-
SANet [3]	67.0	104.5	8.4	13.6	258.4	334.9	-	-	8.2	-
ADCrowdNet (AaD) [17]	70.9	115.2	7.7	**12.9**	273.6	362.0	-	-	**7.3**	-
ADCrowdNet (AbD) [17]	**63.2**	**98.9**	8.2	15.7	266.4	358.0	-	-	7.7	-
SFCN (ImgNet) [32]	-	-	8.9	14.3	-	-	114.8	192.0	-	-
SFCN (GCC) [32]	64.8	107.5	**7.6**	13.0	**214.2**	**318.2**	**102.0**	**171.4**	9.4	-
Ours	**63.3**	**97.8**	**7.5**	**11.5**	**232.3**	**314.8**	**111.1**	**182.8**	**7.1**	9.77

32), fewer congested regions were included, with the increase of window size (*i.e.*, 128), little information would be revealed about sparse regions. Both would compromise the imbalance in training samples. The map generated with window size of 80 is closer to the real congested scenes in the original image. Therefore, window size of 80 is the most plausible option as it is the closest to the congested scenes in the original image.

4.4 Comparison with State-of-the-Arts

To illustrate the effectiveness of MRNet, we conduct a comprehensive comparison of MRNet and nine state-of-the-art crowd counting methods, including ADCrowdNet [17], Cascaded-MTL [26], CL [13], CP-CNN [27], CSRNet [15], MCNN [34], SANet [3], SFCN [32], and Switching-CNN [23]. It is worth noting that ADCrowdNet (AaD) and ADCrowdNet (AbD) are different implementation of ADCrowdNet using linear and binary masks respectively; SFCN (ImgNet) and SFCN (GCC) are different versions of SFCN pretrained on ImageNet and GCC datasets respectively.

Table 5 shows the performance of all methods on four datasets, in which the results of the first place and the second place on each metric are labelled in red and blue, respectively. Due to the unavailability of source codes of most state-of-the-art methods, we use the experiment results provided by works of the authors and other researchers. But none of the methods have been evaluated on the MSE metric in WorldExpo'10 dataset. Through comparison of the performance of MRNet and other methods, we have drawn the following conclusions:

1) In all 9 metrics, MRNet either takes the first or second places (except MSE in WorldExpo'10 dataset). This indicates a clear demonstration of MRNet's superiority over other state-of-the-art crowd counting methods.

2) Our proposed method ranks first on 5 metrics and comes second on 4 metrics. Despite the vastly different common perceptions toward first and second places, it is still a satisfying performance and a much better one than the other methods. It is also worth noting that our proposed method uses common VGG pre-trained weight, and it still outperforms SFCN (GCC), which takes first places on 3 metrics and the second places on 1 metric, and is pretrained on GCC dataset. The GCC dataset consists of 15,212 high-resolution large-scale synthetic crowd images with detailed annotation, which is of great help to improve the accuracy of estimation. Due to the limitation of GPU memory, our proposed method is unable to be pretrained on the same dataset for a fair comparison with SFCN (GCC). If pretrained on the same dataset as MRNet, the estimation accuracy of SFCN degrades significantly (See the comparison between SFCN (ImgNet) and SFCN (GCC) in Table 5), and is outperformed by MRNet on all metrics. This is a clear demonstration of the good generalization ability of our method.

3) When compared with multi-column methods, *e.g.* MCNN, our method achieves better performance on all metrics. The reason could be that the former lacks the ability to handle regions with different congestion degrees, and thus fails to extract features at different scales of the image. Normally, multi-scale methods use single density regressor to generate density map for the entire image. However, as the crowd becomes congested, the scale of person changes, the person's appearance becomes incomplete, and the features of persons in different crowd regions with different congestion degree become difficult for single regressor to learn. MRNet apply multiple density regressors to learn specific features of persons in different congested regions, which reduce the difficulty of learning mapping function and improve the accuracy of estimation in these crowd regions.

4) In comparison with the methods based on deep neural network, such as CSRNet, our proposed method performs better because CSRNet may output the negative density for background scenes and positive density for specific objects other than persons. Our method can filter out the objects other than persons when feature map generated by backbone multiplies the recognition map pixel-wisely. Moreover, with the help of specific regression function, MRNet achieves better performance in estimating density maps with different crowd densities. Figure 8 shows the examples of crowd counting results generated by MRNet on ShanghaiTech, UCF_CC_50, UCF-QNRF and WorldExpo'10 datasets. We can see that our proposed method obtain a robust and state-of-the-art

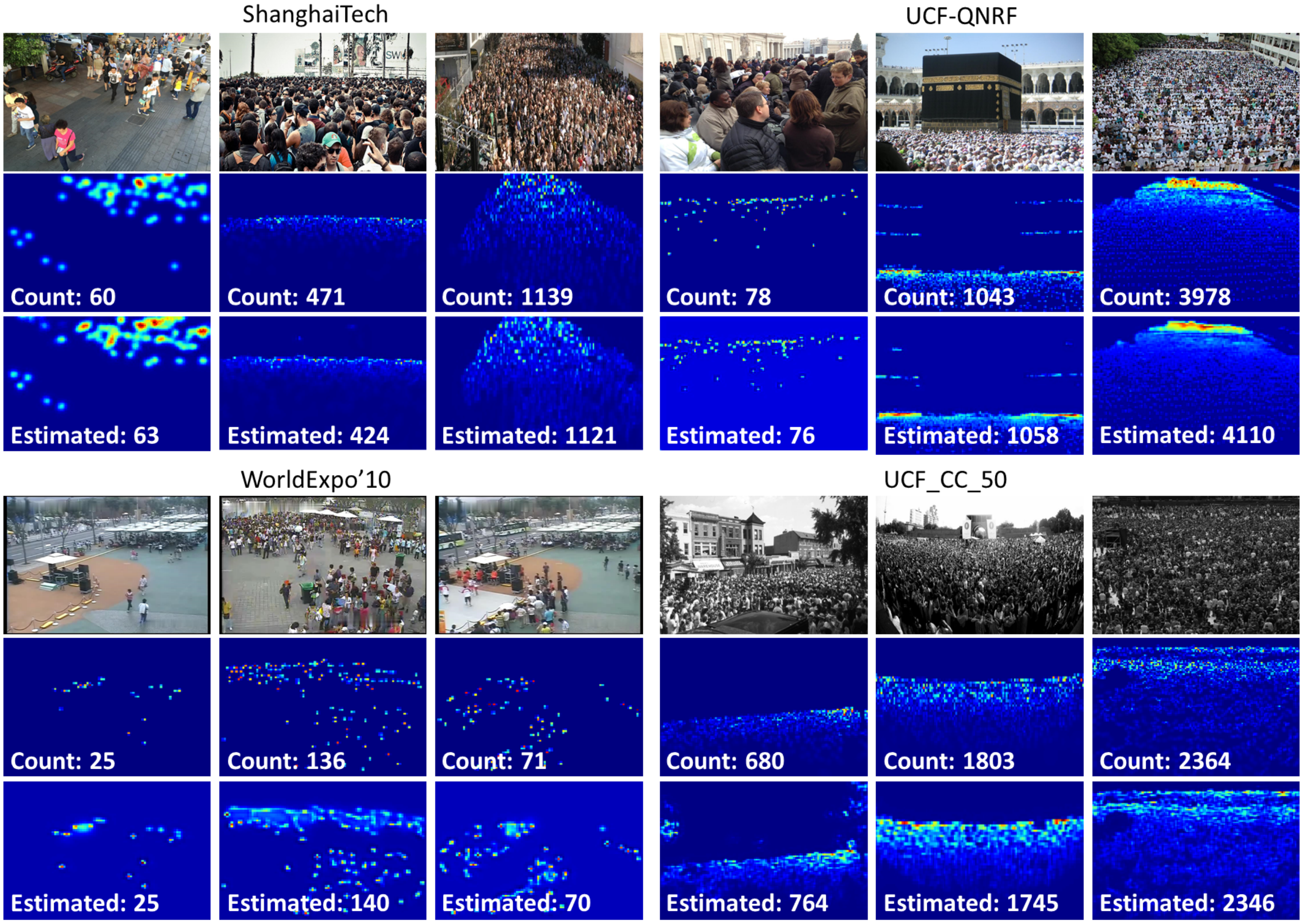

Figure 8: Qualitative examples of crowd counting using MRNet on ShanghaiTech, UCF-QNRF, WorldExpo’10 and UCF_CC_50 datasets. To each dataset, from top to bottom: crowd images, ground truths, and density maps generated by MRNet.

performance on various scenes, and is proved to be equally effective in sparse crowds and congested ones.

5 CONCLUSION

In this work, we proposed a novel multi-layer convolutional neural network called MRNet for identifying crowd scenes with different densities. We used recognition branch to localize crowds in images and extracted the congestion information for regression. Density regressors in each layer obtain specific mapping function for density estimation. Thanks to multi-layer architecture, MRNet is capable of learning different features of images with diverse degrees of congestion, predicting more accurate density maps for both sparse and congested regions. Hence, MRNet is showing more robustness and accuracy in various crowded scenes. On four mainstream crowd counting datasets (ShanghaiTech, UCF_CC_50, WorldExpo’10 and UCF-QNRF), MRNet delivers competitive performance compared with state-of-the-art methods.

ACKNOWLEDGEMENTS

This work is supported by National Science Foundation of China (61202320, 61732007), Science, Technology and Innovation Commission of Shenzhen Municipality (JCYJ20180307151516166), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.

[2] Lokesh Boominathan, Srinivas S. S. Kruthiventi, and R. Venkatesh Babu. 2016. CrowdNet: A Deep Convolutional Network for Dense Crowd Counting. In *ACM International Conference on Multimedia*. 640–644.

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. 2018. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *European Conference on Computer Vision*. 757–773.

[4] Antoni B Chan, Zhangsheng John Liang, and Nuno Vasconcelos. 2008. Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1–7.

[5] Zhang Cong, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. 2015. Cross-Scene Crowd Counting via Deep Convolutional Neural Networks. In *IEEE*

Conference on Computer Vision and Pattern Recognition.

[6] Navneet Dalal and Bill Triggs. 2005. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[7] Anthony C. Davies, Jia Hong Yin, and Sergio A. Velastin. 1995. Crowd Monitoring using Image Processing. *Electronics Communication Engineering Journal* 7, 1 (1995), 37–47.

[8] Jia Deng, Wei Dong, Richard Socher, Lijia Li, Kai Li, and Li Feifei. 2009. ImageNet: A Large-scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition.* 248–255.

[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona. 2012. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 4 (2012), 743.

[10] Pedro F. Felzenszwalb, David McAllester, Deva Ramanan, and Ross B. Girshick. 2010. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 9 (2010), 1627–1645.

[11] Xiaolin Huang, Yuexian Zou, and Yi Wang. 2016. Cost-sensitive Sparse Linear Regression for Crowd Counting with Imbalanced Training Data. In *International Conference on Multimedia and Expo.* 1–6.

[12] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. 2013. Multi-source Multi-scale Counting in Extremely Dense Crowd Images. In *IEEE Conference on Computer Vision and Pattern Recognition.* 2547–2554.

[13] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Almaadeed, Nasir M Rajpoot, and Mubarak Shah. 2018. Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In *European Conference on Computer Vision.* 544–559.

[14] Victor Lempitsky and Andrew Zisserman. 2010. Learning To Count Objects in Images. In *Advances in Neural Information Processing Systems.* 1324–1332.

[15] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition.* 1091–1100.

[16] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. 2018. DecideNet: Counting Varying Density Crowds Through Attention Guided Detection and Density Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition.* 5197–5206.

[17] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. 2019. ADCrowdNet: An Attention-injective Deformable Convolutional Network for Crowd Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition.* 3431–3440.

[19] Daniel Onororubio and Roberto Javier Lopezsastre. 2016. Towards Perspective-Free Object Counting with Deep Learning. In *European Conference on Computer Vision.* 615–629.

[20] Viet Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. 2015. COUNT Forest: CO-voting Uncertain Number of Targets using Random Forest for Crowd Density Estimation. In *IEEE International Conference on Computer Vision.* 3253–3261.

[21] Mengye Ren and Richard S. Zemel. 2017. End-to-End Instance Segmentation with Recurrent Attention. In *IEEE Conference on Computer Vision and Pattern Recognition.* 293–301.

[22] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (2017), 1137–1149.

[23] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. 2017. Switching Convolutional Neural Network for Crowd Counting. In *IEEE Conference on Computer Vision and Pattern Recognition.* 4031–4039.

[24] Chong Shang, Haizhou Ai, and Bo Bai. 2016. End-to-End Crowd Counting via Joint Learning Local and Global Count. In *IEEE International Conference on Image Processing.* 1215–1219.

[25] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations.*

[26] Vishwanath A. Sindagi and Vishal M. Patel. 2017. 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.*

[27] Vishwanath A Sindagi and Vishal M Patel. 2017. Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs. In *IEEE International Conference on Computer Vision.* 1879–1888.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition.* 1–9.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition.* 2818–2826.

[30] Paul Viola and Michael J. Jones. 2004. Robust Real-time Face Detection. *International Journal of Computer Vision* 57, 2 (2004), 137–154.

[31] Elad Walach and Lior Wolf. 2016. Learning to Count with CNN Boosting. In *European Conference on Computer Vision.* 660–676.

[32] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. 2019. Learning from Synthetic Data for Crowd Counting in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[33] Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *International Conference on Learning Representations.*

[34] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-image Crowd Counting via Multi-column Convolutional Neural Network. In *IEEE Conference on Computer Vision and Pattern Recognition.* 589–597.