Density Map Regression Guided Detection Network for RGB-D Crowd Counting and Localization

Dongze Lian[1]*, Jing Li[1]*, Jia Zheng[1], Weixin Luo[1,2], Shenghua Gao[1]†
[1] ShanghaiTech University [2] Yoke Intelligence
{liandz, lijing1, zhengjia, luowx, gaoshh}@shanghaitech.edu.cn

Abstract

To simultaneously estimate head counts and localize heads with bounding boxes, a regression guided detection network (RDNet) is proposed for RGB-D crowd counting. Specifically, to improve the robustness of detection-based approaches for small/tiny heads, we leverage density map to improve the head/non-head classification in detection network where density map serves as the probability of a pixel being a head. A depth-adaptive kernel that considers the variances in head sizes is also introduced to generate high-fidelity density map for more robust density map regression. Further, a depth-aware anchor is designed for better initialization of anchor sizes in detection framework. Then we use the bounding boxes whose sizes are estimated with depth to train our RDNet. The existing RGB-D datasets are too small and not suitable for performance evaluation on data-driven based approaches, we collect a large-scale RGB-D crowd counting dataset. Experiments on both our RGB-D dataset and the MICC RGB-D counting dataset show that our method achieves the best performance for RGB-D crowd counting and localization. Further, our method can be readily extended to RGB image based crowd counting and achieves comparable performance on the ShanghaiTech Part_B dataset for both counting and localization.

1. Introduction

Crowd counting is a task of estimating the number of persons in images or surveillance videos, and it has drawn a lot of attention in computer vision community due to its potential applications in security-related scenarios. Almost all previous works target at RGB image based crowd counting [37, 21, 25, 17] and achieve satisfactory performance on this task. With the popularity of depth sensor, people also propose to study RGB-D crowd counting [36, 1, 4] in surveillance scenarios. Compared with RGB image, depth map provides additional information about the localization of heads [5, 33]. In this paper, we propose to simultaneously count and localize heads with RGB-D data.

Crowd counting methods can be roughly categorized into regression-based approaches and detection-based approaches. Recent works have shown the success of regression-based approaches [37, 18, 21, 25, 17] for density map estimation in crowd counting. However, a crucial issue in these regression-based approaches is that the position of each head is not explicitly given, which restricts the application of regression-based approaches in some related video surveillance tasks, including pedestrian detection [22], anomaly detection [16] and person re-identification [23], *etc.*. In contrast, detection-based crowd counting approaches [29, 30, 31] can provide such head localization information. However, detection-based approaches usually encounter underestimation issues because of the low recall rate for small/tiny heads. Motivated by the success of regression-based approaches as well as advantages of RGB-D data for object detection [5, 33], we propose to leverage density map for more robust detection-based crowd counting with RGB-D data. Next we will analyze the challenges in detection-based approaches, and give our solutions by leveraging RGB-D data.

Challenge 1: Underestimation. Underestimation, which means the number of detected heads is much smaller than the total number of heads (*i.e.* low recall), is a common problem in detection-based approaches. Especially when the heads are small/tiny or occluded, detection-based approaches usually fail to detect them [15]. However, small/tiny heads are very common in real scenarios. For example, about 23% of heads are smaller than 8×8 pixels in ShanghaiTech Part_B as shown in Figure 1 (a).

Our solution. We alleviate this underestimation problem from the following aspects: i) a density map provides a prior about the probability of a pixel being a head. Existing work [37, 12] has shown that the effectiveness of density map estimation for those small/tiny even occluded heads (as shown in Figure 1 (c)), which motivates us to leverage den-

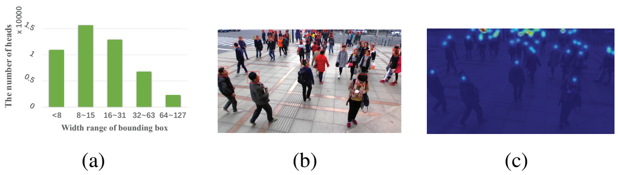
*Equal contribution.
†Corresponding author.

sity map to facilitate the classification branch in detection-based approaches. Thus we propose a regression guided detection network (RDNet) for crowd counting; ii) regression methods would greatly benefit from high-fidelity ground-truth density map in training phase. However, ground-truth density map is usually generated based on a Gaussian kernel with a fixed bandwidth centered at each head without considering the changes of head sizes, whereas such changes can be very significant even within each image, as shown in Figure 1 (b). Obviously such density map generation is not desirable. As depth helps the estimation of head sizes, we propose a depth-adaptive kernel for Gaussian based ground-truth density map generation. Our depth-adaptive kernel generates a high-quality density map for training a more robust regression network, which consequently boosts the performance of detection-based crowd counting; iii) RetinaNet [14] is used for head detection in our paper. One reason for RetinaNet failing to detect small heads is that the anchors are set in higher layers, while for those small/tiny heads, the anchor should be set in lower layers. Luckily, depth provides a prior for estimating the size of heads, which is helpful to determine in which layers we should set anchors as well as the initialization of anchor sizes. We term the strategy of leveraging depth for anchor sizes initialization as depth-aware anchor.

Challenge 2: ground-truth annotation. Detection-based approaches need the annotations of bounding boxes for all heads, but the bounding box annotation is extremely time-consuming compared with point annotation at head center. Besides, occlusion is common in the crowded scenes, such the annotation for those occluded heads with bounding boxes is also much difficult.

Our solution. We propose to estimate size of a bounding box based on depth of the head center, and use the estimated bounding box as ground-truth to train our network. As shown in Figure 1 (b), our estimated bounding boxes can well locate heads. Experiments also show that our strategy achieves state-of-the-art performance for crowd counting and localization.

In view of the importance of RGB-D for detection-based approaches, it is highly demanded to have a large-scale RGB-D crowd counting dataset. However, existing RGB-D dataset is too small [1] for data-driven approaches. Thus we introduce a large-scale RGB-D dataset by capturing images from crowded scenes at different places. Our dataset contains 2,193 images and 144,512 head counts in total. As far as we know, it is the largest RGB-D dataset for crowd counting. In our dataset, each head is annotated with a point at head center, and the bounding box of each head is also provided in test set to facilitate the evaluation of head detection.



(a) (b) (c)

Figure 1. (a) shows the range of bounding boxes width in ShanghaiTech Part_B training data (We generate these bounding boxes with nearest neighbors.). (b) shows the estimated bounding boxes using depth information. (c) is the density map.

Our main contributions are summarized as follows: i) we propose a regression guided detection network (RDNet) for RGB-D crowd counting and localization; ii) depth-adaptive kernel and depth-aware anchor are designed to facilitate density map generation in regression and anchor initialization in detection. We further leverage depth to estimate the bounding box sizes of all heads and use them as the ground-truth to train RDNet; iii) we introduce a large-scale RGB-D crowd counting dataset named ShanghaiTechRGBD for performance evaluation, and such a dataset would accelerate the study of detection-based approaches for crowd counting; iv) our method can be easily extended to RGB image based crowd counting and localization. Extensive experiments validate the effectiveness of our method for both RGB-D and RGB crowd counting.

2. Related Work

2.1. Detection-based Crowd Counting

Early detection-based approaches [20, 29, 30, 31, 11] mainly rely on hand-crafted features, whose performance usually decays seriously for those very crowded scenes with occlusions. Recently, deep learning based approaches have demonstrated their performance for object detection [13, 14]. Thus people attempt to leverage these more advanced detection framework for crowd counting. One example is that Stewart *et al.* [28] proposed an end-to-end people detector for crowded scenes. In very crowd scenes, the head sizes can be extremely small, consequently bounding box annotations may be very difficult sometimes. Thus the ground-truth for crowd counting is usually annotated with a dot at head center, which restricts the exploration of detection-based approaches for crowd counting. Furthermore, most previous objects detection methods cannot well handle the small/tiny objects, which are common in crowd counting. Thus the performance of detection-based approaches are usually inferior to that of regression-based approaches. In this paper, we will show that detection-based approaches can also achieve comparable even better performance by leveraging RGB-D data.

2.2. Regression-based Crowd Counting

Regression-based approaches map an image to its density map where the integration is total number of heads.

Recently, CNN based approaches [37, 18, 21, 25, 17] have shown their advantages over hand-crafted features [10, 3] in learning this nonlinear mapping. According to the change of view angles as well as the change in density at different regions, many networks [37, 21, 25, 12] have been carefully designed and shown their good performance for crowd counting, such as MCNN [37], switch-CNN [21], CSRNet [12], *etc*. We refer readers to a survey paper [26] for more details about CNN based crowd counting. Recently, Liu *et al.* [15] also propose to take advantage of the results of detection for density map regression. In contrast, we leverage regression to improve the detection for crowd counting, and our solution can also provide the location information of heads. To achieve this aim, Idrees *et al.* [7] also propose to simultaneously solve counting, density map regression and localization in recent work. Specifically, their method estimates a binary localization map where head centers correspond to 1's, and all the rest are 0's, but its optimization is not easy, and the estimated locations are coarse due to the downsampling layer in CNN.

2.3. RGB-D Crowd Counting

Although the depth sensors are very popular, only a few works focus on RGB-D crowd counting [32, 1, 36] due to the lack of RGB-D crowd counting dataset. In these works, the depth information was usually used to segment the foreground/background in RGB image or detect the position of head directly. Bondi *et al.* [1] leveraged depth image to help detect the position of head and a RGB-D dataset was proposed in their work. Similarly, Zhang *et al.* [36] proposed an unsupervised water filling method to count people. Song *et al.* [27] utilized the deep region proposal network to perform head detection on the depth images collected by an overhead vertical Kinect sensor. In [4], Fu *et al.* utilized RGB-D information and detected head-shoulder for final crowd counting. However, there are only two RGB-D datasets, and the amount of people is small in both datasets, as shown in Table 1. In this paper, we introduce a large-scale RGB-D dataset, and we leverage depth for designing anchors, generating more accurate ground-truth density maps and estimating bounding boxes for detection-based crowd counting.

3. Method

The overall network architecture of our regression guided detection network (RDNet) for crowd counting is shown in Figure 2. It contains two modules: a density map regression module and a head detection module. In the density map regression module, depth-adaptive kernel is introduced to generate high-fidelity ground-truth density map. In the detection module, we leverage a RetinaNet [14] for detection in view of its advantages in both speed and performance. We feed the estimated density map to the classification branch in the detection network to facilitate the classification of heads, meanwhile, the depth-aware anchor strategy is also proposed to initialize appropriate anchor, which also helps the improvement of detection performance.

3.1. Density Map Regression Module

Density map regression module takes an image as input and leverages CNN for density map estimation. The most commonly used ground-truth density map generation strategy utilizes a Gaussian with fixed bandwidth for approximating the density map. Given a head with location x_i, and if the image contains N heads in total, then the density map of this image can be written as:

$$\mathcal{D}(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_\sigma(x). \tag{1}$$

$G_\sigma(x)$ is a 2D Gaussian kernel with fixed bandwidth σ. Therefore, the crowd counting problem is converted to the following problem: $\mathcal{F} : \mathcal{I}(x) \rightarrow \mathcal{D}(x)$, which learns a mapping from an image space $\mathcal{I}(x)$ to a density map space $\mathcal{D}(x)$. Once the mapping function $\mathcal{F}$ is learnt, the density map of any given image can be obtained and the integration over the whole image is an estimation of total head counts.

A high-fidelity ground-truth density map is desired. Actually, the sizes of heads vary significantly, even for the heads within an image, as shown in Figure 1 (b). Therefore, it is desirable to design different σ's for different heads other than using the same σ for all heads. Bounding box annotation can provide such information, but it is time-consuming than point annotation and it is also hard to annotate bounding boxes for those tiny or occluded heads. In [37], a distance based strategy is used to determine σ for each head, which sets σ linearly proportional to distance between the target head and its nearest neighbors. Such strategy works well for those very crowd areas, while it fails in the areas where people are very sparse. Considering that depth provides information of head sizes within an image under the assumption that all heads are of the same sizes in the real world, we propose a depth-adaptive kernel for density map generation.

As shown in Figure 3, the projection radius of a human head in practice and head in an image are R and r, respectively. f is the focal length of the camera, and d is the depth of head [1]. Because the distance between head and the camera is much larger than the radius of head, we can approximate the diameter of head as $2R$ shown in the Figure 3. According to the camera projection and triangle similarity, we have the following equations:

$$\frac{r}{R} = \frac{s_2}{s_1} = \frac{f}{d} \tag{2}$$

[1] Here, we use a stereo camera and we assume the head radius R is the same. In fact, the height of the camera can slightly impact R. The specific formulation can refer to [6].

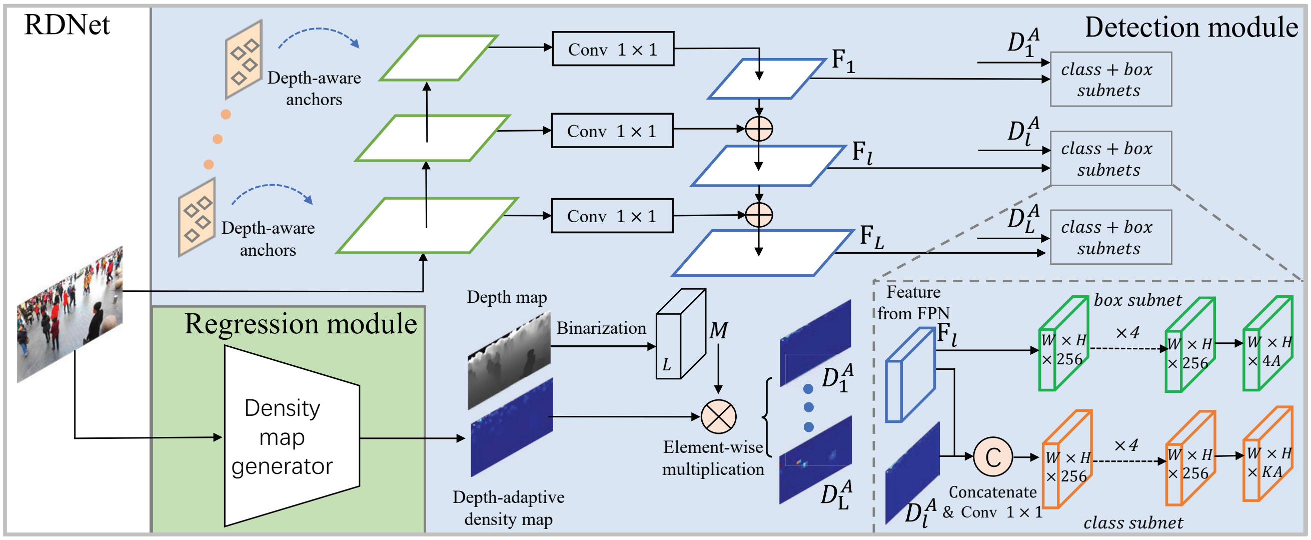


Figure 2. Our RDNet consists of two modules: regression module and detection module.

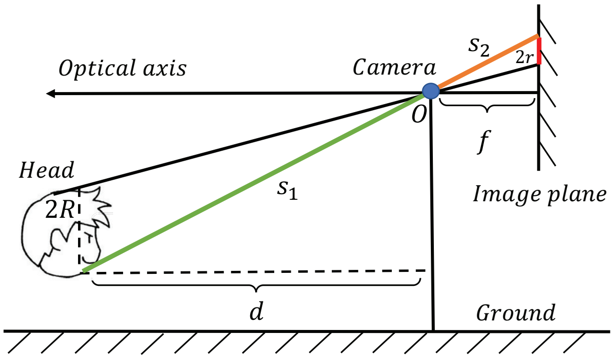


Figure 3. The relationship between the radii of human head in practice and head in image.

and

$$\sigma = \beta r = \beta \frac{Rf}{d} = \beta \frac{\gamma}{d}. \tag{3}$$

Here we let bandwidth σ in Gaussian based density map be proportional with the radius of head in image. We can see that σ is inversely proportional with its depth d for a given head. We generate the density map according to the depths of different head centers and denote such density as the depth-adaptive density map. Specifically, we replace $G_\sigma(x)$ in Eq. (1) with a depth-adaptive Gaussian kernel $G_{\sigma(d)}(x)$ and get depth-adaptive density map $\mathcal{D}^A(x)$.

$$\mathcal{D}^A(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_{\sigma(d_i)}(x). \tag{4}$$

Here d_i corresponds to depth of x_i, and $\sigma(d_i) = \beta \frac{\gamma}{d_i}$. With depth-adaptive density map generation, we employ a CSRNet B [12] (dilation rate = 2) as our regression module in light of its state-of-the-art performance for crowd counting.

3.2. Detection Module

Our detection network is based on a RetinaNet [14] because of its advantages in speed and accuracy. Specifically, RetinaNet is based on a feature pyramid network (FPN) and it contains multi-scale encoding and decoding layers. For each decoding layer, it takes features from corresponding encoding layers as well as outputs from its previous decoding layers as inputs. The detection is conducted on every scale feature map, which includes a class subnet for classifying and a box subnet for regressing bounding boxes.

However, RetinaNet cannot be directly applied for head counting because it fails to detect small/tiny heads, meanwhile crowd counting is only with point based ground-truth annotations rather than bounding boxes. Thus we propose to use estimated density map from regression module and depth-aware anchor to improve the robustness of RetinaNet for small/tiny heads detection and use depth to generate bounding boxes for training RetinaNet.

Density map guided classification. RetinaNet fails to detect those small/tiny heads because the class subnet fails to classify those anchor boxes as positive. However, such class subnet would benefit from density map. Density map shows the distribution of heads, and its value at each pixel is related to the probability of the pixel being a head. Therefore we propose to feed the estimated density map into the detection network to boost the performance of small/tiny heads. RetinaNet detects heads of different scales at different decoding layers. The lower layers respond to the detection of smaller heads, and higher layers respond to the detection of larger heads. We thus propose to mask density map based on the depth map. Specifically, for a given decoding layer l (l=1,…,L), suppose the sizes of heads to be detected in this layer is between $[r_1, r_2]$, based on Eq. 3, we can estimate the depth of the heads $d \in [\frac{\gamma}{r_2}, \frac{\gamma}{r_1}]$. Then we generate a binarization matrix $M \in B^{L \times H_d \times W_d}$ based on depth map, where H_d and W_d are the height and width of generated density map, respectively. In binarization, the depth map is downsampled to the same size as the density map. For each channel l in M, the values of pixels with larger or smaller corresponding depth are set to 0's, and the values of pixels within the range are 1's. We denote this binary mask as M_l, and use it to mask our estimated density map:

$$D_l^A = D^A \odot M_l \tag{5}$$

where $\odot$ means the element-wise multiplication, and D_l^A is the masked density map corresponding to l^{th} layer. Then we simply concatenate this masked density map with features F_l from the l^{th} decoding layers to help heads/non-heads classification (as shown in Figure 2). In the specific implementation, we choose $L = 5$, which means extracting 5 scale feature maps for classification and regression.

Depth-aware anchor. One reason for the general detector failing to detect small/tiny heads directly is that the anchors are set in higher layers, while for those small/tiny heads, the anchor should be set in lower layers. With depth information, we can get a prior for estimating the size of heads, which is helpful to determine which layer we should set anchors as well as the initialization of anchor sizes. We term the strategy of leveraging depth for anchor initialization as depth-aware anchor. Our depth-aware anchor not only reduces the search space [35], as well as facilitates the initialization of anchor sizes. We follow the Eq. 3 to generate depth anchor: $H(m,n) = \frac{\gamma}{d(m,n)}$, where (m,n) is the index of height and width in depth-aware anchor map, and $d(m,n)$ is the corresponding depth at this location.

Generation of bounding box for training. Define bounding boxes set $\mathcal{B} = \{b_1, ..., b_N\}$ for N heads. According to the Eq. 3, we can estimate width w_i of b_i as follows: $w_i = \dfrac{\gamma}{d_i}$, where we assume that γ is the same for all images [2]. We set the bounding boxes as squares ($w_i = h_i$). For those position with invalid depth value, we employ nearest neighbor to generate bounding box according to [37].

3.3. Loss Function

For regression module, we adopt the Euclidean distance to measure the distance between the estimated density map and the ground-truth. The loss function can be defined as:

$$L_R(\Theta) = \frac{1}{2\mathcal{M}} \sum_{k=1}^{\mathcal{M}} \left\| E_k(\mathcal{I}_k; \Theta) - \mathcal{D}_k^A \right\|_2^2 \tag{6}$$

where Θ are the CNN model parameters to be learned. The $\mathcal{I}_k$ is the k-th training image and $\mathcal{M}$ is the total number of training images. The $\mathcal{D}_K^A$ is the ground-truth depth-adaptive density map and E_K is the density map estimated by the regression module. The $L_R(\Theta)$ is the loss between the estimated density map and the ground-truth density map.

For detection module, detection loss consists of classification loss and bounding box regression loss as follows:

$$L_D = L_{cls} + \lambda L_{reg} \tag{7}$$

where

$$L_{reg}(p) = \begin{cases} 0.5(p)^2, & \text{if } |p| \leq 1 \\ |p| - 0.5, & \text{otherwise} \end{cases} \tag{8}$$

and λ is the weight to balance classification loss and bounding box regression loss. We firstly optimize the regression module according to the Eq. 7, and train the detection module. Finally we fine-tune the whole network.

We implement our proposed method with the PyTorch [19] framework. We empirically choose $\lambda = \dfrac{1}{9}$, $\gamma = 5$ in our implementation. The size of input image is 540×960 for efficiency. We conduct our experiments on NVIDIA Titan X Maxwell GPU with batch size 4 and learning rate 10^{-4}, respectively. We train and test on each dataset independently and Adam [9] optimizer is employed. Following RetinaNet [14], we only randomly flip images horizontally for data augmentation.

4. Experiments

4.1. Evaluation Metrics

We follow the standard evaluation metrics for crowd counting evaluation [37, 21]: mean absolute error (MAE) and mean square error (MSE).

$$\text{MAE} = \frac{1}{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} \left| N_j - \hat{N}_j \right|, \tag{9}$$

$$\text{MSE} = \sqrt{\frac{1}{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} \left(N_j - \hat{N}_j \right)^2} \tag{10}$$

where $\mathcal{M}$ is the number of the test images, N_j and $\hat{N}_j$ represent ground-truth and estimated number of heads in the j^{th} test image, respectively. The estimated number of heads $\hat{N}_j$ is the total number of all the detection bounding boxes.

In addition, to evaluate the detection performance of RDNet, we manually label the bounding boxes in the test set of our RGB-D dataset as ground-truth. We follow the standard binary Average Precision (AP) calculation method and classify a sample as positive one if $IOU > 0.5$ between the bounding boxes of ground-truth and prediction. For those images only with point annotations, we evaluate the localization performance and calculate Average Precision (AP)

[2] Actually γ is slightly related to angle and height of the camera, here we just ignore their effect.

Table 1. Comparisons of ShanghaiTechRGBD with some existing datasets: **Num** is the number of images; **Max** is the maximal crowd count within one image; **Min** is the minimal crowd count; **Ave** is the average crowd count; **Total** is total number of labeled heads.

Dataset		Resolution	Num	Max	Min	Ave	Total	Modality
CBSR [36]	Dataset 1	240 × 320	2834	7	0	1.6	4,541	Depth
	Dataset 2	240 × 320	1500	7	0	1	1,553	RGB + depth
MICC [1]		480 × 640	3358	11	0	5.32	17,630	RGB + depth
ShanghaiTechRGBD		1080 × 1920	2193	234	6	65.9	144,512	RGB + depth

according to [7]. Following [7], we classify whether a predicted head point is positive example or not based on the following criteria:

$$\text{Prediction} = \begin{cases} \text{Positive}, & \text{if dist} \leq \theta \\ \text{Negative}, & \text{otherwise} \end{cases} \tag{11}$$

where dist is the distance between the predicted head point and ground-truth. We calculate AP by varying the threshold θ. In order to distinguish between AP for detection and AP for localization, we denote AP for detection as AP_det, and AP for localization as AP_loc.

4.2. Evaluations on RGB-D Crowd Counting Datasets

4.2.1 Datasets

ShanghaiTechRGBD. To facilitate the performance evaluation of data-driven approaches for crowd counting, we introduce a large-scale RGB-D dataset named ShanghaiTechRGBD that contains 2,193 images with 144,512 annotated head counts. The images in ShanghaiTechRGBD are captured by a stereo camera (ZED[3]) whose valid depth ranges from 0 to 20 meters. The scenes in our dataset include busy streets of metropolitan areas and crowded public parks. The lighting condition ranges from very bright to very dark in different scenarios. Some representative images in ShanghaiTechRGBD are shown in Figure 4. The histograms of crowd counts and the statistics of heads with different depth in ShanghaiTechRGBD are shown in Figure 5. We also compare ShanghaiTechRGBD with other RGB-D crowd counting datasets in Table 1, and we can see that ShanghaiTechRGBD is the most challenging RGB-D crowd counting dataset in terms of the number of images and heads. We randomly choose 1,193 images as training set and use the remaining as test set.

The MICC dataset. The MICC dataset is introduced by [1]. It is acquired by a surveillance camera in indoor scenes. There are three video sequences in the MICC dataset: FLOW, QUEUE and GROUPS. In the FLOW sequence, people walk from one point to another, while in the QUEUE sequence, people move slowly in line. In the GROUPS sequence, people do not move out of a controlled area. It should be noticed that the participants are of the same for these three sequences. There are 3,542 head counts in 1,260 frames in FLOW sequence, 5,031 head counts in 918 frames in QUEUE sequence and 9,057 head counts in 1,180 frames. The ground-truths are annotated with bounding boxes. Previous work [1] leverages unsupervised learning on MICC. Here we use 20% images of each scene in this dataset as training set and use the remaining as test set.

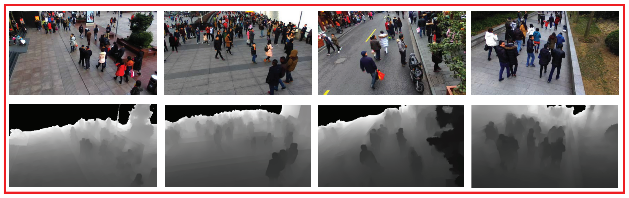
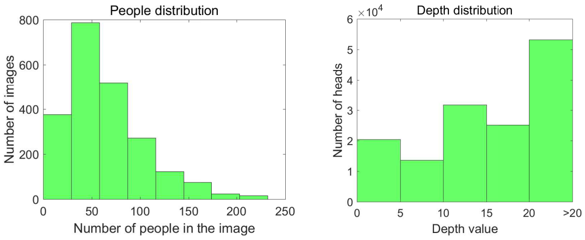


Figure 4. Some images on the ShanghaiTechRGBD dataset.



(a) (b)

Figure 5. (a) Statistics histogram of ShanghaiTechRGBD. (b) Depth distribution (values over 20 m are invalid.).

4.2.2 Performance Comparison

To evaluate the effectiveness of our method, we conduct experiments on the ShanghaiTechRGBD dataset and MICC dataset with some state-of-the-art methods. i) MCNN [37]. MCNN leverages a multi-column convolution network with kernels of different sizes to count heads with different sizes; ii) MCNN-adaptive. We replace the density map generated by Gaussian with fixed bandwidth with our depth-adaptive kernel in MCNN; iii) CSRNet [12]. CSRNet leverages a dilated CNN to expand the reception field. It achieves the state-of-the-art performance on many datasets; iv) CSRNet-adaptive: We replace the density map generated by Gaussian with fixed bandwidth with our depth-adaptive kernel

[3] https://www.stereolabs.com/

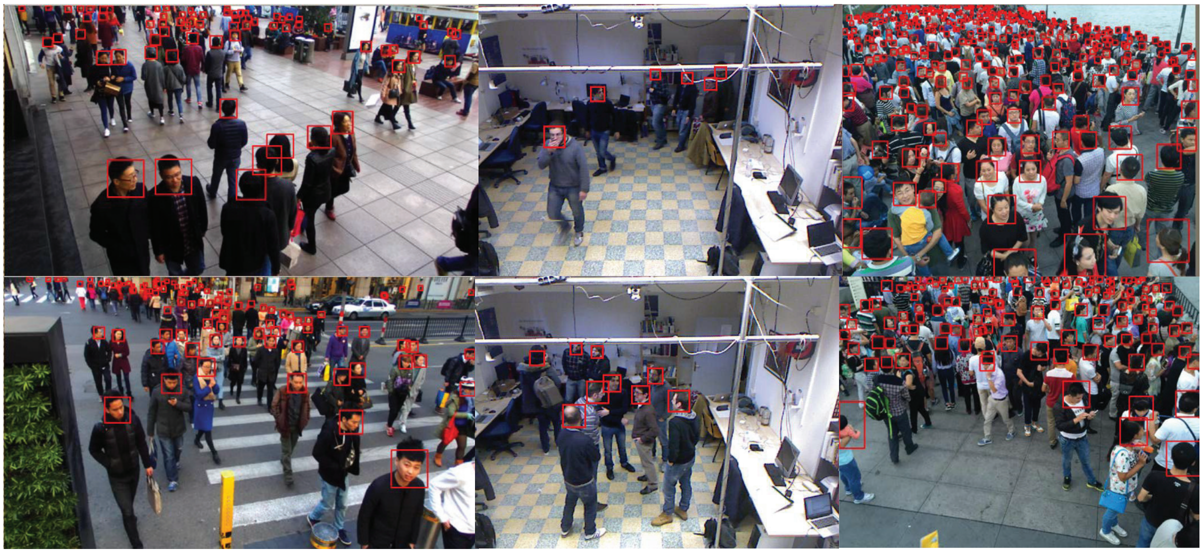


Figure 6. The detection results of RDNet on ShanghaiTechRGBD, MICC and ShanghaiTech Part_B are from left to right, respectively. More detection results and failure cases are shown in supplementary material.

in CSRNet; v) DecideNet (DetNet) [15]. DecideNet leverages the results of detection for density map estimation; vi) Idrees *et al.* [8]. Idrees *et al.* design the compositional loss to estimate density map, localization map and head counts; vii) RetinaNet. Here, we use the estimated bounding boxes with depth information to train the RetinaNet; viii) RetinaNet*. As comparison, we also use the bounding boxes of the fixed size to train the RetinaNet, and the size is fixed as average sizes of all estimated bounding boxes with depth information.

Table 2. Performance evaluations on ShanghaiTechRGBD.

Methods	MAE	MSE	AP_det
MCNN [37]	7.56	10.92	-
MCNN-adaptive	7.14	9.99	-
CSRNet [12]	5.11	7.34	-
CSRNet-adaptive	**4.91**	**7.11**	-
RetinaNet [14]	10.25	14.56	0.356
RetinaNet* [14]	21.84	36.19	0.136
DecideNet (DetNet) [15]	9.74	13.14	0.383
Idrees *et al.* [8]	7.32	10.48	-
RDNet	**4.96**	**7.22**	**0.610**

The performance of different methods are shown in Table 2 and Table 3. We can see that our RDNet achieves the best performance compared with detection-based methods and comparable performance compared with regression-based methods. Further, we have the following observations: i) depth-adaptive kernel always outperforms Gaussian kernel with fixed bandwidth for all regression-based approaches, which validates its effectiveness for ground-truth generation; ii) the results of RetinaNet are not satisfactory because of the underestimation problem, as shown in Figure 7. While with the help of density map and depth-aware anchor, our method greatly improves head counting, especially for those small/tiny heads; iii) the improvement of RetinaNet over RetinaNet* validates the effectiveness of our bounding box estimation strategy for training RDNet.

Table 3. Performance evaluations on the MICC dataset.

Methods	MAE	MSE	AP_det
MCNN [37]	1.500	2.259	-
MCNN-adaptive	1.489	2.114	-
CSRNet [12]	1.359	2.125	-
CSRNet-adaptive	**1.343**	**2.007**	-
RetinaNet [14]	1.641	2.554	0.476
DecideNet (DetNet) [15]	1.541	**2.382**	0.481
Idrees *et al.* [8]	1.396	2.642	-
RDNet	**1.380**	2.551	**0.505**

To evaluate the accuracy of localization, we compare our method with Idrees *et al.* [8] in terms of AP_loc metric in Figure 8. We can see that our method always achieves better AP_loc than Idrees *et al.* in the three datasets, which validates the effectiveness of our method for localization.

4.3. Ablation Studies

To understand the effectiveness of different modules in our RDNet, we conduct ablation studies, as shown in Table 4. It is worth noting that our method is a detection-based method. As shown in the last three rows in Table 4, depth-adaptive kernel (DAK) and depth-aware anchor (DAA) help detect and count heads. Meanwhile, the first two rows in Table 4 show the effectiveness of DAK for the regression-

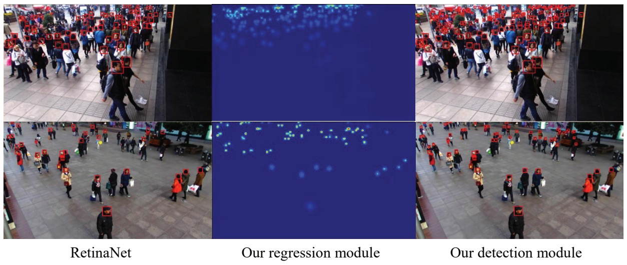


Figure 7. From left to right: the detection results from RetinaNet on ShanghaiTechRGBD. The density map regression results from our regression module. The detection results from our RDNet. We can find our method can detect small/tiny heads than RetinaNet.

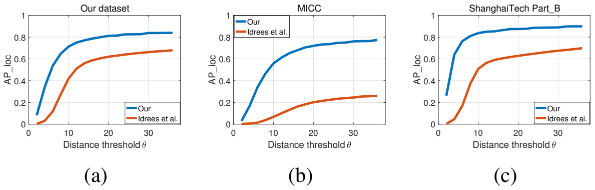


Figure 8. (a), (b), (c) are AP_loc comparisons on our RGB-D dataset, MICC and ShanghaiTech Part_B, respectively.

based method, where DAA is not applicable. The DAK improves the performance of regression-based crowd counting, and the DAA facilitates the detection.

Table 4. Ablation studies on our dataset (DAA: depth-aware anchor; DAK: depth-adaptive kernel.).

	DAK	DAA	**MAE**	**MSE**	**AP_det**
Reg	×	N/A	5.11	7.34	N/A
	√	N/A	**4.91**	**7.11**	N/A
Det	×	×	5.64	8.04	0.593
	×	√	5.31	7.54	0.604
Our	√	√	**4.96**	**7.22**	**0.610**

4.4. Evaluation on RGB Crowd Counting Dataset

Our RDNet can be easily extended to RGB image crowd counting by removing the depth-aware anchor and depth-adaptive kernel. Since there is no depth, we only simply feed the density map in layer one without masking into the class subnet. We evaluate the performance of RDNet on ShanghaiTech Part_B [37] for RGB based crowd counting. Similar to our dataset, ShanghaiTech Part_B is also a dataset with surveillance view. Here, we use the bounding boxes estimated by nearest neighbors strategy [37] as ground-truth to train RDNet due to the lack of depth.

We compare our method with other state-of-the-art methods on ShanghaiTech Part_B in Table 5. We can see that our method achieves the comparable performance with some regression-based methods. It is worth noting that we only use coarse bounding boxes of heads as supervision in the training phase due to the lack of bounding box annotations and depth, the performance can be further improved if the bounding box are provided. We choose CSRNet [12] as regression module, the improvement of our method over CSRNet validates the effectiveness regression guided detection strategy. Further, by comparing the performance of improvement of SANet over CSRNet (2.2 MAE), and the improvement of our method over CSRNet (about 1.8 in terms of MAE), we can see that our model would probably benefit from better regression module, such as SANet.

We also compare our method with Idrees *et al.* [8] in terms of localization metric AP_loc, and show the results in Figure 8 (c). Our method achieves higher localization precision. Some bounding boxes prediction results are shown in Figure 6. We can see that our method can locate the heads accurately, even for those small ones.

Table 5. Evaluation results on the ShanghaiTech Part_B dataset.

Methods	**MAE**	**MSE**
Zhang *et al.* [34]	32.0	49.8
MCNN [37]	26.4	41.3
Cascaded-MTL [24]	20.0	31.1
Switch-CNN [21]	21.6	33.4
CP-CNN [25]	20.1	30.1
SANet [2]	**8.4**	**13.6**
DecideNet (DetNet) [15]	44.90	73.18
Idrees *et al.* [8]	15.5	24.9
CSRNet [12]	10.6	16.0
Ours	**8.8**	**15.3**

5. Conclusion

A regression guided detection network (RDNet) is proposed for RGB-D crowd counting and localization, which leverages a density map to boost the performance of detection for crowd counting. With the help of depth, i) a depth adaptive kernel is designed, which generates high-fidelity ground-truth density map and facilitates the regression-based crowd counting; ii) a depth-aware anchor is designed. Our depth-aware anchor facilitates the anchor initialization, and improves the detection of small heads; iii) even with point annotations, we can still use depth to estimate the sizes of bounding boxes, which shows their effectiveness for training RDNet. We further collect the large-scale ShanghaiTechRGBD crowd counting dataset for performance evaluation. Experiments on our dataset and MICC show that our method achieves the best performance for RGB-D crowd counting. Further, our method can be extended to RGB crowd counting and achieves comparable performance on the ShanghaiTech Part_B dataset.

Acknowledgement. We want to thank Desen Zhou, Yingying Zhang, Siqin Chen for their help in collecting data.

References

[1] Enrico Bondi, Lorenzo Seidenari, Andrew D Bagdanov, and Alberto Del Bimbo. Real-time people counting from depth imagery of crowded environments. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 337–342. IEEE, 2014.

[2] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[3] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 545–551. IEEE, 2009.

[4] Huiyuan Fu, Huadong Ma, and Hongtian Xiao. Real-time accurate crowd counting based on rgb-d information. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 2685–2688. IEEE, 2012.

[5] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014.

[6] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.

[7] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018.

[8] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *The European Conference on Computer Vision (ECCV)*, September 2018.

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

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

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

[12] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, 2018.

[13] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017.

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

[16] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6536–6545, 2018.

[17] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7661–7669, 2018.

[18] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[20] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *2011 International Conference on Computer Vision*, pages 2423–2430. IEEE, 2011.

[21] D Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 6, 2017.

[22] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[23] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Person re-identification with deep similarity-guided graph neural network. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[24] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017.

[25] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1879–1888. IEEE, 2017.

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

[27] Diping Song, Yu Qiao, and Alessandro Corbetta. Depth driven people counting using deep region proposal network. In *Information and Automation (ICIA), 2017 IEEE International Conference on*, pages 416–421. IEEE, 2017.

[28] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.

[29] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *null*, page 734. IEEE, 2003.

[30] Meng Wang and Xiaogang Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3401–3408. IEEE, 2011.

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

[32] Mingliang Xu, Zhaoyang Ge, Xiaoheng Jiang, Gaoge Cui, Bing Zhou, Changsheng Xu, et al. Depth information guided crowd counting for complex crowd scenes. *Pattern Recognition Letters*, 2019.

[33] Xiangyang Xu, Yuncheng Li, Gangshan Wu, and Jiebo Luo. Multi-modal deep feature learning for rgb-d object detection. *Pattern Recognition*, 72:300–313, 2017.

[34] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 833–841. IEEE, 2015.

[35] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018.

[36] Xucong Zhang, Junjie Yan, Shikun Feng, Zhen Lei, Dong Yi, and Stan Z Li. Water filling: Unsupervised people counting via vertical kinect sensor. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 215–220. IEEE, 2012.

[37] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.