JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DIFFERENT CNN MODELS FOR CROWD COUNTING

[1]Rishabh Jajoriya, [2]Pranav Kumar Singh
[1]Software Engineer, [2]Software Engineer
[1]Software Engineering,
[1]Delhi Technological University, Rohini Delhi, India

Abstract : The three models analyzed in this study - MCNN, CSRNet, and LSC-CNN - are all CNN-based methods that directly estimate the crowd density map from an input image. MCNN uses multiple column networks to capture different scales of crowd information, while CSRNet employs a dilated convolutional neural network to increase the receptive field and better handle scale variations. LSC-CNN, on the other hand, introduces a local self-correction mechanism to reduce the effects of occlusion and background clutter. To evaluate the performance of these models, the study uses three widely used evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Peak Signal-to-Noise Ratio (PSNR). The outputs tells us that all three models are better than traditional regression-derived methods and best performance is achieved to date on benchmark datasets. However, there are significant differences in their computational efficiency and robustness to variations in crowd density and occlusion. MCNN is the most computationally efficient model but struggles with varying densities and occlusion, while CSRNet is more robust to these factors but has higher computational requirements. LSC-CNN combines the advantages of both models by using a local self-correction mechanism to improve robustness and achieving comparable performance to CSRNet. The study also explores some possible directions for future research, such as combining multiple models or using attention mechanisms to better handle occlusion. Overall, accurate crowd counting is a critical task with numerous practical applications, and CNN-based density map estimation methods have shown significant improvements in recent years. But as we know that there is always possibility for betterment considering efficiency, robustness, and generalization to different scenarios. As such, this study offers valuable insights for researchers and professionals seeking to improve crowd counting techniques and their applications in various domains. By systematically evaluating the strengths and limitations of different models, this study provides a foundation for future developments in this field.

IndexTerms **– MCNN, CSRNet, LSC-CNN.**

1 INTRODUCTION

Analyzing crowds quickly is important for public safety and planning, but it's a challenging Computer Vision task. Crowd counting has become a prominent research area due to its relevance in various fields. As a result, many works have been published to calculate the amount of objects in photos or videos from different fields, including crowd counting [1-4], vehicles [2], leaves [6]. Crowd counting is essential for safety in events with large gatherings. As the world population grows and urbanization increases, effective crowd counting techniques are in demand. Crowd counting has emerged as an important application of computer vision, and CNN-based models have demonstrated strong performance in this task. Our paper presents an efficient crowd counting model by comparing it with others.

2 RELATED WORKS AND SCOPE

Estimating crowd count has various methods which can be classified into : detection, regression, density estimation, and CNN based density estimation methods. This article mainly focuses on CNN-based density estimation, while also assessing relevant works in other areas for completeness. In the initial stages of crowd counting [7-8], detection-based techniques were used, such as detecting a person's head via sliding window. Although object detectors like R-CNN [9-11], YOLO [12], and SSD [13] achieve good accuracy in sparse settings, they struggle with occlusion and background clutter in dense crowds. To address these issues, regression-based methods have been developed that use global or local features and regression techniques such as linear or Gaussian mixture regression to learn a mapping function for crowd counting. Recently, CNN-based density estimation methods such as MCNN [1] and CSRNet [3] have achieved significant accuracy improvements by directly estimating the density map of a crowd from an input image. Furthermore, LSC-CNN [14] has been proposed to improve the performance of CNN-based methods by using a local self-correction mechanism to mitigate the effects of occlusion and background clutter.

3 DIFFERENT NETWORK ARCHITECTURE FOR CROWD COUNTING

3.1 Multi-column Convolutional Neural Network (MCNN)

MCNN is a crowd counting method that estimates the crowd count from a single photo, regardless of density or perspective, using a Multi-column CNN architecture. The network can adjust according to difference in people/head size and input image size/resolution. The density map is made accurately using geometry-adaptive kernels without requiring a perspective map. The proposed model was evaluated on a dataset of 716 photos, with 330k heads tagged, and outperformed all known methods. Additionally, the model can be easily transferred to other datasets.

3.1.1 Density map based crowd counting

To calculate the number of heads in an image using CNN these are the choices available: first is a network with an image input and the result is the crowd count, and second is generating a crowd density map and computing the head count through integration. This study supports the second option for specific reasons.

a. The density map is a valuable resource as it provides more detailed information compared to the overall crowd count. It displays the location arrangement of the all the people in the image, which can be useful in various applications. For instance, if there is a higher density in a particular region compared to other areas, it may suggest that there is something unusual happening in that location.
b. CNN-generated density maps are more adaptable to heads of varying sizes, making them suitable for arbitrary inputs with varying perspective effects. This improves crowd counting accuracy as the filters are more semantically relevant.

3.1.2 Density map through geometry-adaptive kernels

CNN must learn to calculate density of crowd from input images, making the condition of training data crucial. Here's how to convert an image of labelled human heads into a crowd density map. Suppose there is a person present at pixel x_i, we represent it as a delta function $\delta(x\text{-}x_i)$. As a result, a picture with N labelled heads may be shown as a function.

$$H(x) = \sum_{i=1}^{N} \delta\,(x - x_i) \tag{1}$$

To convert a discrete crowd count function $F(x)$ to a continuous density function G_σ we can intertwine it with a Gaussian kernel [15], such that the density is $F(x) = H(x) * G_\sigma(x)$. However, this density function is based on the assumption that x_i represents samples which are not dependent and are present on the picture plane, that is false as each x_i is a sample of the ground cover density in the 3D image, and the associated pixels with various x_i correspond to regions that differ in size. To account for perspective distortion while estimating crowd density, we can assume an even distribution around each head and estimate the distortion utilizing the mean distance separating each head from its k closest neighbouring heads. In crowded scenarios, it is difficult to determine head size and its relationship with the density map due to occlusion. However, we have found that in crowded scenarios, the distance between two nearby individuals' centers is often proportional to head size (see figure 1). For density maps in crowded scenarios, we suggest estimating the spread parameter for each individual based on their mean distance to neighbours.[1]

We denote the distances from each head x_i in an image provided to its k nearest neighbours as $\{d_1^i, d_2^i, \ldots, d_m^i\}$. The average distance of 12mm is computed $\bar{d}^i = \frac{1}{m}\sum_{j=1}^{m} d_j^i$. Therefore, the pixel that is related to x_i is consistent with a region on the ground in the picture with a radius scaled in proportion to $\bar{d}^i$. To compute the density of individuals surrounding pixel x_i, we need to intertwine $\delta(x\text{-}x_i)$ with a Gaussian kernel with deviation σ_i scaled in accordance with $\bar{d}^i$. To be more specific, the density F should be

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_{\sigma_i}(x), with\ \sigma_i = \beta \bar{d}^i \tag{2}$$

[1] We immediately use the supplied density maps or density maps generated from perspective maps in our experiments for pictures given density or perspective maps. Since the available data consists of only a few people with similar head sizes, we use a fixed spread parameter for all of them.

or a given parameter β. The labels H are convolved using geometry-adaptive kernels that adjust to the local geometry. β = 0.3 produced the best results. See Figure 1 for examples of density maps from our dataset.

Fig. 1. Images with their respective crowd density map [1].

3.1.3 Multi-column CNN for density map estimation

To capture crowd density at different scales, a Multi-column CNN (MCNN) with kernels of various sizes for each column is used. The MCNN has three parallel CNNs with varying-sized local receptive fields as filters, and The activation function used for the 2 × 2 area under max pooling is ReLU.

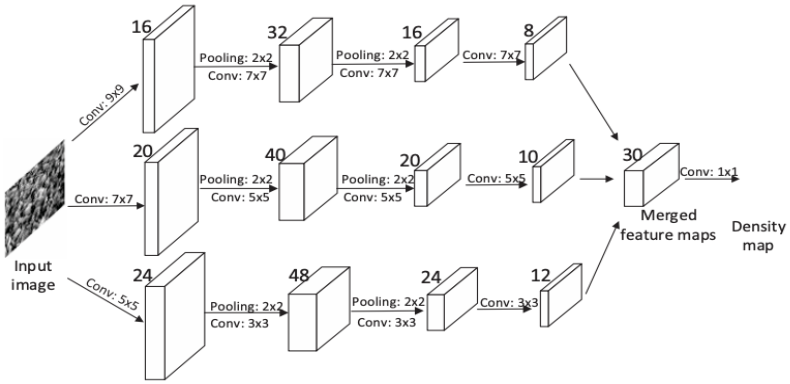

Fig. 2. The suggested multi-column CNN's topology for estimating crowd density maps [1].

To map the density map to a stack of all the CNNs output feature maps, filters with sizes of 1 × 1 are utilized [1]. These feature maps are then mapped to the density map using the same filters. Euclidean distance is used to measure the variation between the density map which was estimated and the actual density map. The following defines the loss function:

$$L(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \| F(X_i;\Theta) - F_i \|_2^2, \tag{3}$$

In the MCNN model, Θ represents the set of parameters that can be learned through training. The training process for this model involves utilizing a collection of N images to optimize the model parameters, denoted by X_i , where F_i is the corresponding ground truth density map. $F(X_i;\Theta)$ is the estimated density map produced by the MCNN for a given input image X_i and a set of parameters Θ. The difference between the predicted density map and the actual density map is measured by the loss function denoted as L.

3.2 Congested Scene Recognition Network (CSRNet)

We improved CSRNet by removing the fully-connected layers of the VGG-16 front-end and using only convolutional layers. However, this makes generating high-quality density maps challenging due to the reduced output size. To address this, we used dilated convolutional layers [3] in the back-end to maintain the output resolution while gathering deeper saliency information. Utilizing VGG-16 in the front-end and dilated convolutional layers in the back-end led to enhanced accuracy of the density map compared to previous designs.

3.2.1 Dilated convolution

The differentiated convolutional layer is a fundamental component of our architecture. A two-dimensional dilated convolution is defined as follows:

$$y(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i, n + r \times j)w(i,j) \tag{4}$$

$y(m,n)$ is the result of dilated convolution on the input $x(m,n)$ and a filter $w(i,j)$ using M and N as the length and breadth, respectively. The dilation rate is represented by the parameter r. Dilated convolution [3] is a type of convolutional layer that improves segmentation accuracy without adding parameters or operations, using a dilation rate represented by the parameter r. This layer is a good alternative to pooling layers that can cause spatial resolution loss. Dilated convolution allows for multi-scale contextual information aggregation while maintaining resolution and outperforms pooling + deconvolution in preserving feature map resolution.

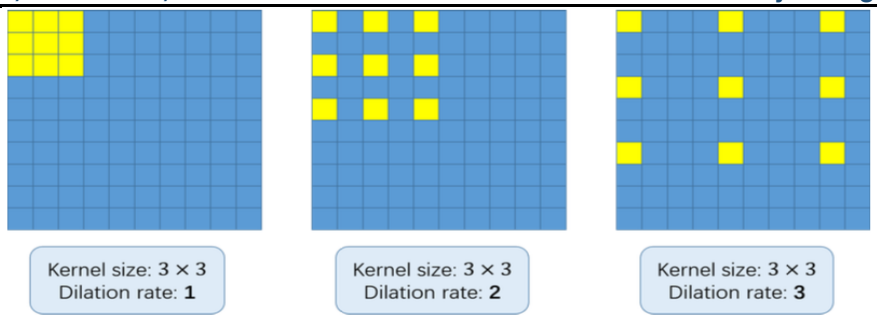

Fig. 3. Design with dilation rates of 1, 2, and 3 [3].

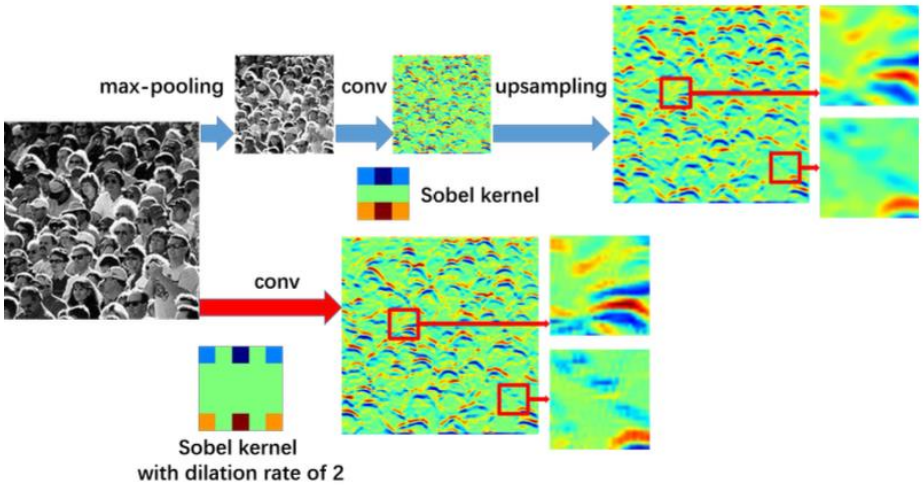

Fig. 4. Comparison is made between dilated convolution and pooling, convolution, and upsampling processes. The Sobel kernel of size 3 × 3 with dilation rate of 2 is used in both approaches [3].

3.2.2 Training details

The CSRNet is trained by fine-tuning the first ten convolutional layers of a pre-trained VGG-16 with a Gaussian initialization for the other layers. Stochastic gradient descent (SGD) is used with a constant rate of learning of $1e-6$. The following is the loss function:

$$L(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \| Z(X_i;\Theta) - Z_i^{GT} \|_2^2, \quad (5)$$

In the formula, N represents the batch size used in the training process, and $Z(X_i;\Theta)$ represents the result of the CSRNet model using the indicated settings in Θ for the input image X_i. The input image is represented by X_i while Z_i^{GT} represents the ground truth result of the input image X_i.

3.3 Locate, Size and Count (LSC-CNN)

The new approach proposed for dense crowd counting aims to locate individuals in crowded areas and count them by sizing their heads with bounding boxes. This approach replaces the traditional density regression method and poses unique challenges due to the high diversity in dense crowds. To overcome these challenges, the LSC-CNN model is designed with a parallel architecture with multiple branches and top-down feature modulation. The training methodology proposed in this study can be executed with only point-based annotations, yet it estimates approximate head size information, resulting in improved performance in head localization and counting when compared to previous density regressors.

3.3.1 Our Approach

A crowd counting dense detection model named LSC-CNN was developed, that predicts head boxes instead of using density regression. The model has three parts: Feature Extractor, Top-down Feature Modulators, and Non-Maximum Suppression [14]. The Feature Extractor extracts features, the TFM networks predict boxes, and NMS selects valid detections to generate the final output.

3.3.1.1 Feature Extractor

The performance of CNN object detectors is heavily dependent on the deep feature extractor network used. To extract features at various resolutions, we use VGG-16 based networks with modified and repeated blocks. The first five convolutional blocks of VGG-16 [14] are used as the backbone network, which is replicated to generate feature maps at different scales. This enables each scale branch to specialize in capturing dense crowds. The network takes as input an RGB image of a crowd, which has a resolution of $\mathbf{224 \times 224}$, and the output is downsampled due to max-pooling. To avoid any conflicts, the replicated blocks are initialized with copied weights. Please refer to Figure 5 for further details.

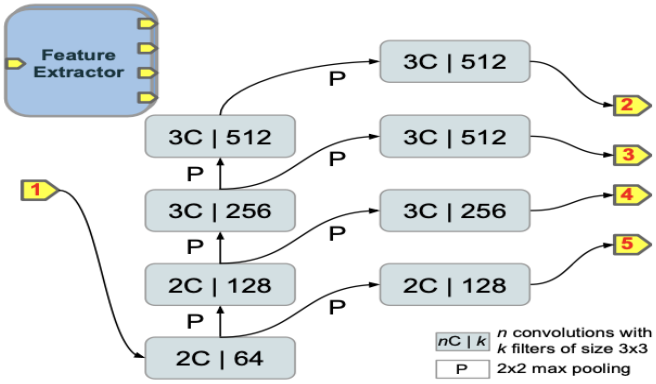

Fig. 5. The precise setup a customised class of VGG-16's Feature extractor that generates feature maps at several sizes [14].

3.3.1.2 Top-down Feature Modulator

The TFM module is a component of the Crowd Counting Network, which is a person detector operating at different scales. It utilizes multi-scale feature processing to provide information on a global context and reduce erroneous detections. The module has four terminals: Terminal 1 is furnished with a set of scaled features from the feature extractor and passes it via a 3×3 filter layer with m set to half of the incoming scale branch. The features extracted from the top-down path are outputted from Terminal 2 to be used as input for the next TFM module, while Terminal 3 accepts these top-down feature maps and uses two convolution layers for top-down processing. The resulting feature maps are concatenated and input into a set of 3×3 convolution layers by reducing the number of filters to provide with the final answer. Terminal 4 provides the result, which categorizes every pixel into one of the preset boundary boxes for the identified head, using a softmax nonlinearity. The output maps are used to build per-pixel confidence levels, with the $1 + n_B$ classes, where n_B is the no. of specified classes boxes. n_B is a hyper-parameter that regulates the fineness of the sizes and is commonly set to three, for a total of $n_S \times n_B = 12$ boxes for all the branches. The first channel of the prediction for scale s, D_{0s} is for background and the remaining $\{D_{1s}, D_{2s}, \ldots, D_{ns}\}$ maps are for the boxes [14].

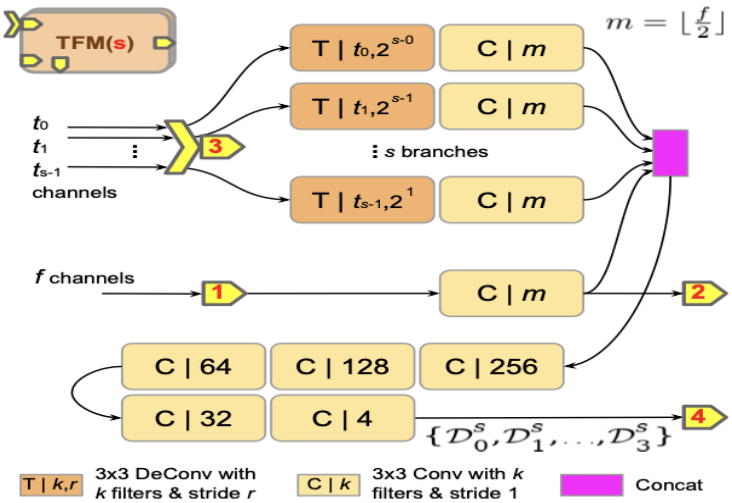

Fig. 6. The Implementation of Top-down Feature Modulator module [14].

3.3.2 Size heads

3.3.2.1 Box classification

LSC-CNN utilizes TFM Modules to locate persons and assign bounding boxes to their heads using preset box sizes instead of regressing box parameters. The model determines the classification of each head based on the per-pixel confidence of the box classes. Ground truth head sizes are difficult to annotate for dense crowd datasets, so point annotations are used to approximate the head sizes and generate a fake ground truth for training. Box sizes are discretized to predefined bins, with finer-sized boxes for high resolution branches and coarser boxes for low resolution branches. This per-pixel box classification approach effectively addresses these challenges while overcoming point annotation difficulties.

Fig. 7. Ground truth samples derived from pseudo boxes. Boxes of the same hue all belong to the same scale branch [14].

4 COMPARING THE PERFORMANCE OF THE MODELS

4.1 Evaluation metric

To assess the performance of the models described in section 3, we evaluate them using the following metrics:

4.1.1 Mean Absolute Error (MAE)

This metric measures the accuracy of the estimates and is used in our evaluation of the models described in section 3.

$$MAE = \frac{1}{N}\sum_{1}^{N}|z_i - \hat{z}_i| \tag{6}$$

N refers to the no. of images used for testing, z_i refers to the no. of people who are actually present in the ith image, and $\hat{z}_i$ refers to the no. of people estimated in the ith image.

4.1.2 Mean Square Error (MSE)

This metric is used to assess the robustness of the estimates.

$$MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(z_i - \hat{z}_i)^2} \tag{7}$$

N refers to the no. of images used for testing, z_i refers to the no. of people who are actually present in the ith image, and $\hat{z}_i$ refers to the no. of people estimated in the ith image.

4.1.3 Peak Signal-to-Noise Ratio (PSNR)

This metric indicates the quality of the output density map.

$$PSNR = 10\log_{10}({}^{255^2}/_{MSE}) \tag{8}$$

4.2 Our Dataset

We have used a dataset which contains 716 images of dense crowds of people of Shanghai. It contains almost 330k heads annotated. Out of the 716 images 400 ($\mathbf{55.8}\%$) were used for training the models (section 3) and rest which is 316 ($\mathbf{44.1}\%$) were used for testing and estimating crowd count.

4.3 Results

Table 1. Comparing performance of different models

Model	MAE	MSE	PSNR
MCNN	29.3	47.5	31.3
CSRNet	34.3	57.1	30.5
LSC-CNN	7.9	12.6	37.0

From Table 1 we can see that LSC-CNN [14] model has the lowest MAE and MSE among the other models and from that we infer that it is the most efficient model for crowd counting among others discussed in section 3 as lower values of MAE or MSE indicate that the model provides more accurate and good quality estimates as well.

5 CONCLUSION

In this paper we have done comparison between three models namely MCNN, CSRNet, LSC-CNN as discussed in section 3 to accurately estimate crowd count. We worked on a dataset which contains a total of 330,165 persons annotated, to better test the performances of crowd counting methods under actual situations. Studies show that these models outperforms existing regression approaches when it comes to crowd counts. Among these models LSC-CNN provides the most efficient estimate of crowd count. Considering this, we expect that the community would abandon the existing regression technique in favour of the more practical dense detection. Further studies might resolve false detections and improve head sizing accuracy.

REFERENCES

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, Single-image crowd counting via multi-column convolutional neural network, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 589-597. IEEE, Las Vegas, NV, USA (2016).

[2] D. Onoro-Rubio, R. J. López-Sastre, Towards perspective-free object counting with deep learning, In European Conference on Computer Vision (ECCV), pp. 615-629. Springer, Amsterdam, The Netherlands (2016).

[3] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, In Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1091-1100. IEEE, Salt Lake City, UT, USA (2018).

[4] R. R. Varior, B. Shuai, J. Tighe, D. Modolo, Scale-aware attention network for crowd counting, In Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, California, USA (2019).

[5] E. Walach, L. Wolf, Learning to count with cnn boosting, In European Conference on Computer Vision (ECCV), pp. 660-676. Springer, Amsterdam, The Netherlands (2016).

[6] S. Aich, I. Stavness, Leaf counting with deep convolutional and deconvolutional networks, In International Conference on Computer Vision (ICCV), pp. 2080-2089. ICCV, Venice, Italy (2017).

[7] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, In CVPR, vol. 1. pp. 878-885. IEEE, San Diego, CA, USA (2005).

[8] M. Enzweiler, D. M. Gavrila, Monocular pedestrian detection: Survey and experiments, In TPAMI, vol. 31, no. 12, pp. 2179-2195. IEEE, (2009).

[9] R. Girshick, Fast r-cnn, In ICCV, pp. 1440-1448. IEEE, Santiago, Chile (2015).

[10] S. Ren, K. He, R. Girshick, J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks, In NIPS, pp. 91-99. NIPS, Montreal Convention Center, Montreal, Canada (2015).

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, Mask r-cnn, In ICCV, pp. 2961-2969. IEEE, Venice, Italy (2017).

[12] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, In CVPR, pp. 779-788. IEEE, Las Vegas, NV, USA (2016).

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in ECCV, pp. 21-37. Springer, Amsterdarm, The Netherlands (2016).

[14] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, R. V. Babu, Locate, Size, and Count: Accurately Resolving People in Dense Crowds via Detection, In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 8, pp. 2739-2751. IEEE, (2021).

[15] V. Lempitsky, A. Zisserman, Learning to count objects in images, In Neural Information Processing Systems (NIPS), pp. 1324-1332. NIPS, Hyatt Regency, Vancouver CANADA (2010).