Crowd Counting and Density Estimation in Highly occluded Complex Scenarios

By Mayamin Hamid Raha
Computer Science and Engineering Department
University of Nevada, Reno
Reno, United States
https://orcid.org/0000-0002-5674-9016

Introduction Crowd counting and Density estimation requires estimating the number of people in congested public scenarios along with identifying the regions of interest (ROIs) having most densely connected number of people which is a possible source of crowd anomalies. It is an important area of research for public security and comes with a lot of challenges associated with occlusion, complex background scaling, high noise, low illumination and non-uniform distribution.

With the increase of world population, events associated with human mob turbulance is rising. There is an overwhelming need of crowd monitoring as the incidences of riot, unlawful gathering of mass people resulting in chaos keep soaring. Surveillance approaches these days requires humans to manually detect criminal activities from videos which is both human resource consuming and prone to errors. So, automating violence detection is the need of the hour and automating crowd counting is an essential part of it. In our research, we accomplished the following.

- Build a CSRNet model architecture for crowd counting and corresponding density map generation.
- Experiment with network architecture and parameter setup to observe the MAE, MSE ,PSNR and IOU scores on part A of Shanghaitech dataset. [1].

I. Related Work

One of the fundamental tasks of crowd analysis is to find densely crowded regions. In the past, computer vision researchers were very interested in object detection and in [2] Lempitsky et al. put forward an interactive object counting method using ridge regression method to estimate object density. Many works followed [2], and then the focus of the researchers shifted to people counting. They started with Head-shoulder detection to estimate number of people in image or video [3]. Gradually, many works have been done on people counting, such as [1], [4]–[9]. Yet, changes of background and crowd density are yet to be overcome. For instance, Idress et al. in their crowd counting approach [9] used SIFT and Fourier analysis to count the number of people in extremely concentrated crowd images.Here the rate of false positives increases with less number of people and temporal information is lost as it is only limited to single images. Zhang et al. [4] used a deep learning approach for estimating crowd density and count.Although this performs well for many datasets, it has a dependency on perspective maps.

Figure 1 : Images from ShanghaiTech Dataset

A. Traditional Crowd Counting

Going into the depth of crowd counting method, We see that we can divide crowd counting methods in two main categories namely, detection based, regression based and density estimation based approaches according to [10]. Regression based approaches solved occlusion and complex background problem to some extent but didnot consider the spatial information of image. Moreover, crowd counting can be categorized into direct methods and indirect methods [11]. Direct method has 2 sub classes one dealing with all features of human as a whole [12] and the other method is related to training a classifier with extracted features. An instance of the latter type can be found in [13]. The most common problem with traditional methods is that it works well only for less crowded scenarios and fails to classify highly occluded or dense scenarios properly. On the other hand indirect method learns a mapping from image to the total number of people in corresponding density map [14]. In these approaches, the problem of speed remained and it would not work in real time for counting people from public surveillance videos.

B. CNN based Crowd Counting

Due to it's ability of learning non-linear features, CNNs have shown immense success in crowd counting and crowd density estimation. After seeing the remarkable performance of CNN in image classification [15], researchers became interested in applying it on Regions of Interest (ROI) for people counting [16]. With increasing complex scenarios of

Identify applicable funding agency here. If none, delete this.

crowd and use of CNNs there are still a lot of parameters and computational cost is very high. CNN based crowd counting can also be divided into two categories based on whether it is dealing with images [17] or with videos(where temporal information has much importance) [18]. Furthermore, on basis of training dataset we can divide crowd counting based on CNN into 3 subcategories these are mentioned below.

subcaption

1) Detection based approach: This method can accurately locate objects and count them [19].However, approach like this doesnot work very well for scenarios where there is multiple overlapping between people. To solve this, Li et al. [20] used Recurrent neural network (RNN) to avoid predicting the same target more than once.

2) Regression based approach: Regression based CNN method can be divided into 2 categories namely: Regression counting and Regression density map.The only difference among them is that regression counting can directly give the people count, whereas, regression density map can give crowd count along with the density map.Generally in higher dense scenarios the latter performs better than the former.

3) Detection and Regression combined: Here both are combined for improved counting and positioning [21].This performs better in scenarios where locating people in dense scenes is very tough.

C. Some of the Challenges faced by different types of Deep learning models for crowd counting

1) Multi-scale model: Most of these crowd counting method extract features through different convolution kernels where there is no interaction between different convolution sub networks. This leads to poor performance and is addressed by Shen et al. [22] in their paper where they have used U-Net to estimate image at pixel level.

2) Context-aware model: It deals with taking the context into account by taking whole image as input and directly generating a high quality density map as output [23].Currently, most CNN networks use limited contextual information and only look for crowd features ignoring individual features that lead to wrong classification. This is solved in [24] by using a combination of top down and bottom up network to associate high dimensional context information with low dimensional features of CNN regression and predict crowd density map respectively.

3) Auxiliary-task model: Marsden et al. [25] proposed a Resnet18 based architecture that can do crowd counting, density estimation and violent event detection in auxiliary task manner so that their performance improves on each task (each task mutually helping one another).In high density areas, these methods are still quite inactive and can be improved by enhanced resolution [23]

4) Lack of labeled data: To deal with very little data with labels, researchers have put forward three different types of models. The first one is Semi-supervised model, where a combination of supervised and unsupervised features are used and performs well in classification [26]. again for highly crowded scene, existing point annotation datasets donot work very well. To solve this, Lei et al. in [27] used multiple auxiliary task to effectively predict the density of weakly annotated images which proved to be very effective as a weakly supervised model.The third type is self supervised and it deals with both limited data and overfitting problem. To this end, Liu et al. in their paper [6] used self supervised auxiliary task to improve the counting task performance. He introduced a method to rank unlabeled datasets based on sub-image and resizing them based on number of people that can be present in each patch.

5) Domain adaptation model: Due to diverse environment scenarios many exsiting models donot perform well in highly dense scenarios. In order to address this, Wang et al. [8] in introduced a system of generating synthetic data and labeling them without manual intervention.

6) Perspective map model: To get a better local optimization Zhang et al. [] used two different loss functions for crowd counting and density estimation and optimized them alternately which lead to better performance.They also proposed to generate density map based on perspective changes that helped to increase robustness of network for scene scaling.

7) Attention mechanism model: The idea is to fuse features of different scales to solve scale change of crowd counting. A common challenge here is the decreasing counting accuracy in scenarios that are noisy and very congested. It was solved in [28] by adding attention mechanism and multi-scale deformable convolution to their network named Attention-injective Deformable Convolutional network (ADCrowdNet).

8) Network architecture search model: Previously hand-crafted density estimation networks were used for crowd counting which lacked diversity and prone to error. [29] used a neural network architecture search to make an autonomous crowd counting model. it searches for encoder decoder network and applies network architecture search for crowd counting tasks.

In our proposed method, we aim to use state of the art CSRNet [30] for generating a crowd counting model that works both on dense and sparsely crowded scenes.To this we end we will be using Shanghaitech Dataset part A and part B that contains both dense and sparsely crowded scenarios respectively.

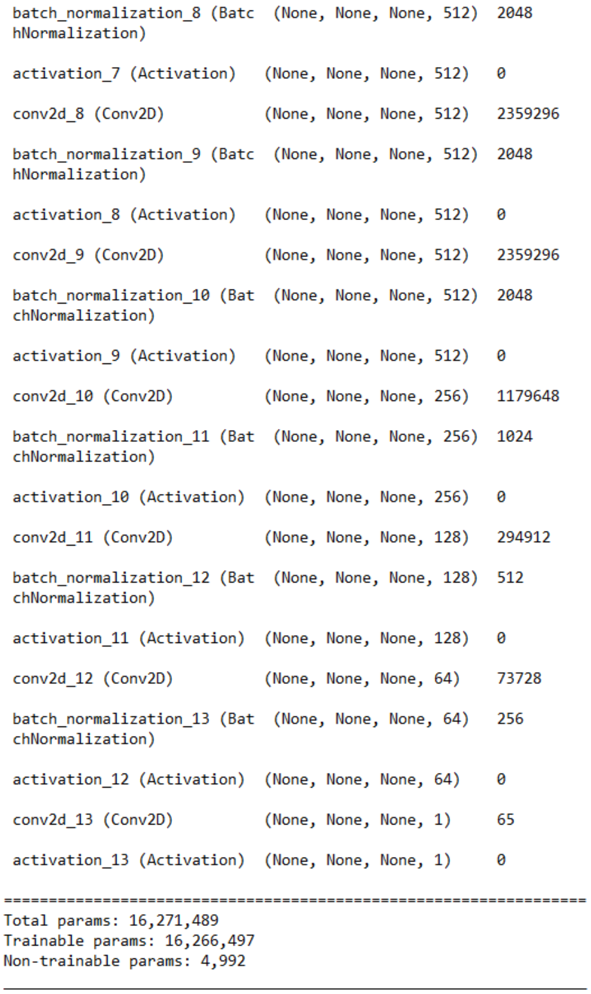
II. Architecture

Out of 4 different variations of CSR-Net lite we build the one that has the following structure.Here, we had 16,271,489 toal number of parameter out of which 16,266,497 are trainable and 4,992 are not trainable.

III. Methodology

A. Architecture

- Front-end: We select VGG 16 [31]as the front-end of CSRNet because of its efficient transfer learning ability. Here we remove the classification part of VGG-16 (fully-connected layers) and build the proposed CSRNet [30] with convolutional layers in VGG-16.The output



```
batch_normalization_8 (Batc  (None, None, None, 512)  2048
hNormalization)

activation_7 (Activation)   (None, None, None, 512)  0

conv2d_8 (Conv2D)           (None, None, None, 512)  2359296

batch_normalization_9 (Batc  (None, None, None, 512)  2048
hNormalization)

activation_8 (Activation)   (None, None, None, 512)  0

conv2d_9 (Conv2D)           (None, None, None, 512)  2359296

batch_normalization_10 (Bat  (None, None, None, 512)  2048
chNormalization)

activation_9 (Activation)   (None, None, None, 512)  0

conv2d_10 (Conv2D)          (None, None, None, 256)  1179648

batch_normalization_11 (Bat  (None, None, None, 256)  1024
chNormalization)

activation_10 (Activation)  (None, None, None, 256)  0

conv2d_11 (Conv2D)          (None, None, None, 128)  294912

batch_normalization_12 (Bat  (None, None, None, 128)  512
chNormalization)

activation_11 (Activation)  (None, None, None, 128)  0

conv2d_12 (Conv2D)          (None, None, None, 64)   73728

batch_normalization_13 (Bat  (None, None, None, 64)   256
chNormalization)

activation_12 (Activation)  (None, None, None, 64)   0

conv2d_13 (Conv2D)          (None, None, None, 1)    65

activation_13 (Activation)  (None, None, None, 1)    0

=================================================================
Total params: 16,271,489
Trainable params: 16,266,497
Non-trainable params: 4,992
_________________________________________________________________
```

Figure: Our Baseline Network Architecture

size of the front-end network will be 1/8 of the original input size. Here we will use only 3 x 3 kernels
-end: While we keep on stacking convolutional and pooling layers of VGG 16 our ouput size may shrink too much due to which we will use dilated convolution as a back-end for extracting deeper information with out additional computational complexity.

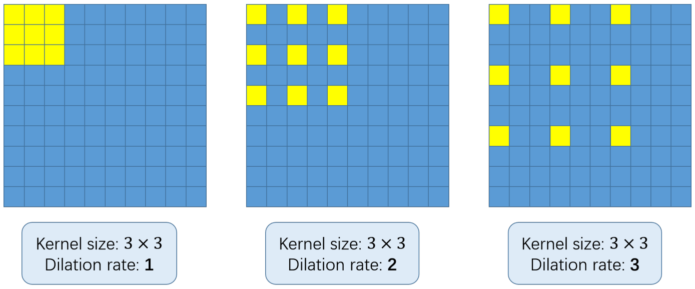
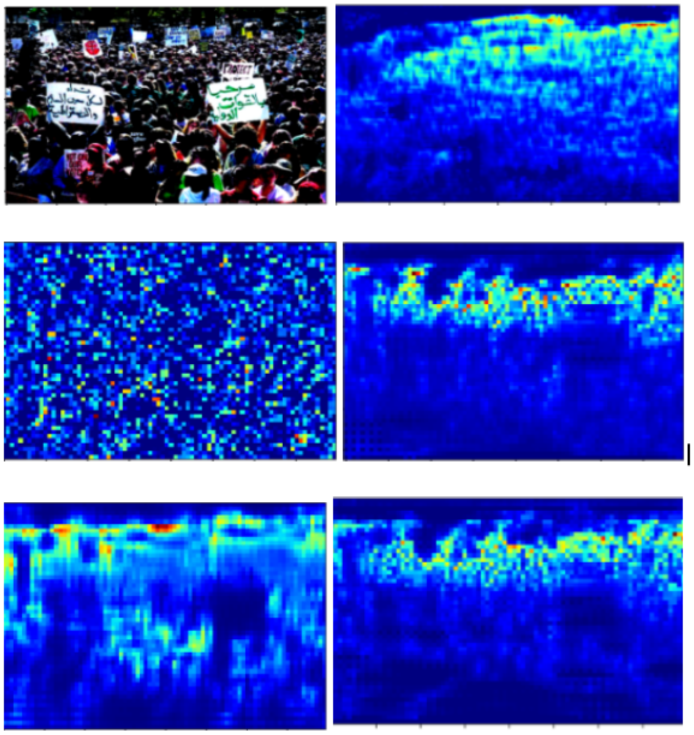


Figure: Visualization of Dialated Convolutions

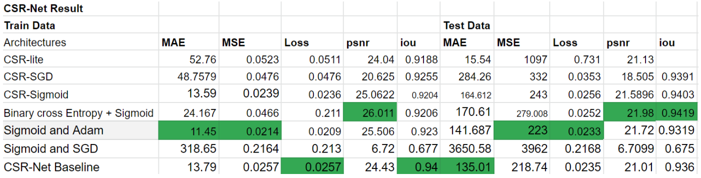
- Training For training we will use first 10 convolutional layers are fine-tuned from a well-trained VGG-16.For the other layers, the initial values come from a Gaussian initialization with 0.01 standard deviation. Stochastic gradient descent (SGD) will be applied with fixed learning rate at 1e-6 for training. Also, we choose the Euclidean distance to measure the difference between the ground truth and prediction.We will use the loss function used by [30]



Density map generated by CSR-Net(Baseline), CSRNetLite, CSR-Net with Sigmoid, CSR-Net with Binary Cross Entropy, CSR-Net with SGD, CSR-Net with Sigmoid and SGD

Figure: Density maps generated by our models

- Data augmentation At first we will crop 9 patches from each image at different locations with 1/4 size of the original image. The first four patches will contain four quarters of the image without overlapping while the other five patches will be randomly cropped from the input image. After that, we will mirror the patches so that we double the training set.



CSR-Net Result										
Train Data						Test Data				
Architectures	MAE	MSE	Loss	psnr	iou	MAE	MSE	Loss	psnr	iou
CSR-lite	52.76	0.0523	0.0511	24.04	0.9188	15.54	1097	0.731	21.13	
CSR-SGD	48.7579	0.0476	0.0476	20.625	0.9255	284.26	332	0.0353	18.505	0.9391
CSR-Sigmoid	13.59	0.0239	0.0236	25.0622	0.9204	164.612	243	0.0256	21.5896	0.9403
Binary cross Entropy + Sigmoid	24.167	0.0466	0.211	26.011	0.9206	170.61	279.008	0.0252	21.98	0.9419
Sigmoid and Adam	11.45	0.0214	0.0209	25.506	0.923	141.687	223	0.0233	21.72	0.9319
Sigmoid and SGD	318.65	0.2164	0.213	6.72	0.677	3650.58	3962	0.2168	6.7099	0.675
CSR-Net Baseline	13.79	0.0257	0.0257	24.43	0.94	135.01	218.74	0.0235	21.01	0.936

Figure: Results of our experiments with Baseline built by us

B. Experiment

All the deep learning models were run in local runtime using colab notebook and tensorflow keras libraries.The computer had Intel(R) Core(TM) i9-990KF CPU

@ 360GHz 3.60 GHz with 32 GB RAM having Windows 10 Pro operating system.For baseline reconstruction,We took idea from ipnyb notebook implementation of CSR-Net https://github.com/RTalha/CROWD-COUNTING-USING-CSRNET. We collected the dataset from Kaggle and had to modify the code, debug it as well as found faults in some data and removed the redundant ones.

We performed 6 types of experiment after recreating the current baseline model given by the authors known as CSR-Net. Here the metrics used for evaluation are Mean Average Error (**MAE**), Mean squared error (**MSE**), Intersection over Union (**IoU**) and Peak Signal to Noise Ratio (**PSNR**). The results of all the experiments along with the performance of the baseline on part A of given dataset is visualized in Figure

At first, we decrease the depth of each layer of the model. We call it CSR-Net lite having **2 layers of 3x3 kernel having 128 filters each, followed by pairs of 3x3 convolution layers having depths of 64 and 16 respectively**.

Secondly, we changed the loss function to Binary Cross Entropy along with changing of optimizer to **SGD**.

For our third experiment, we change the final layer activation of CSR-Net to **Sigmoid** from RELU and it performs very well.

For our fourth experiment, we used **Sigmoid** as the activation function of final layer and used **Binary-cross entropy** as loss function

Apart from these, we also used adam optimizer along with sigmoid in final activation layer which also showed promising results.

For our final experiment we use **SGD** in place of optimizer and **Sigmoid** activation function in the final layer of the network.

IV. Results

For deep learning tasks associated with crowd counting and density estimation the metrics used for evaluation of people count are MAE, MSE and for evaluation of quality of Density Map PSNR(peak signal to noise ratio) are generally used. The authors of CSR-Net used these metrics along with these we also used IoU and loss metrics for better understanding each of our results. We compare our different experiments with the baseline model of the author.In the paper, the way the authors calcualted MAE and MSE is different from ours due to which out table doesnot match with the ones in the paper, but for relative comparison we buld the baseline first , recorded it's result and then compare that with the MAE and MSE result of our experimental models.

From the results we see that the density maps generated by CSR-Net with Binary cross Entropy and Sigmoid activation functions for test and train data have a better PSNR score(the more the better it is). Moreover, on training CSR-Net with Sigmoid activation function along with Adam loss function performs well in people counting in training dataset with MAE and MSE score of 11.45 and 0.0214 (the less the better it is). The baseline CSRNET model reconstructed by us has a better MAE score of 135.01 on test data compared to all other experimental setups. Here, we also see that decreasing the network layers by half had an adverse affect on the performance of CSR lite it has an MAE of 1097 on test data and 52.76 on train data. Since the source of the data for shanghaitech that we used here had some issues with their given ground truth and test data hence seeing the results of the experiment we can conclude that CSRNet with Sigmoid and Adam along with Sigmoid and Binary cross entropy has much potential as CSRNet for crowd counting and density map generation. We also show some of the note-worthy loss graphs for some of our experiments.

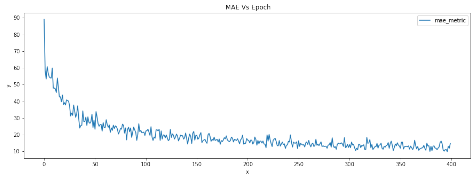


Fig. 1. MAE of CSR-Net with Sigmoid and Binary crossEntropy

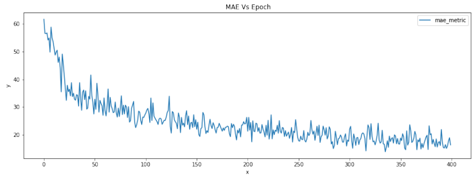


Fig. 2. MAE of CSRlite

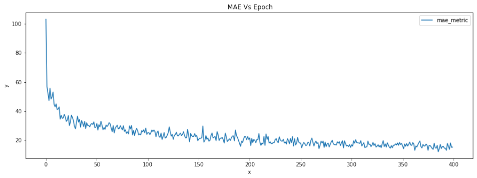


Fig. 3. MAE of CSR-Net with Sigmoid

V. Conclusion

We conclude that CSR Net and experiments based on it has a lot of potential in crowd counting and density map generation. The results we show are for part A of Shanghai tech dataset which contains densely packed images of people.Due to time constraint and resource limitation we were not able to test the performance of our experimental models on part B that contains sparsely crowded images. At some point, CSRNet with sigmoid in final layer performs very well sometimes better than baseline model. It will be interesting to see how this performs on more modern datset like UCF50, UCF Qnrf and crowd images from Vis drone challenge.

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*, 2016, pp. 589–597.

[2] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *European conference on computer vision*. Springer, 2014, pp. 504–518.

[3] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *2008 19th international conference on pattern recognition*. IEEE, 2008, pp. 1–4.

[4] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 833–841.

[5] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento, "A method for counting people in crowded scenes," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2010, pp. 225–232.

[6] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7661–7669.

[7] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.

[8] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 8198–8207.

[9] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2547–2554.

[10] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, simulation and visual analysis of crowds*. Springer, 2013, pp. 347–382.

[11] Z. Fan, H. Zhang, Z. Zhang, G. Lu, Y. Zhang, and Y. Wang, "A survey of crowd counting and density estimation based on convolutional neural network," *Neurocomputing*, vol. 472, pp. 224–251, 2022.

[12] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 878–885.

[13] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Decision forests for computer vision and medical image analysis*. Springer, 2013, pp. 143–157.

[14] A. B. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 545–551.

[15] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1299–1302.

[16] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1215–1219.

[17] X. Ding, Z. Lin, F. He, Y. Wang, and Y. Huang, "A deeply-recursive convolutional network for crowd counting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1942–1946.

[18] Z. Zou, H. Shao, X. Qu, W. Wei, and P. Zhou, "Enhanced 3d convolutional networks for crowd counting," *arXiv preprint arXiv:1908.04121*, 2019.

[19] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2325–2333.

[20] W. Li, H. Li, Q. Wu, F. Meng, L. Xu, and K. N. Ngan, "Headnet: An end-to-end adaptive relational network for head detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 482–494, 2019.

[21] J. Liu, C. Gao, D. Meng, and A. 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*, 2018, pp. 5197–5206.

[22] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5245–5254.

[23] S. Amirgholipour, X. He, W. Jia, D. Wang, and M. Zeibots, "A-ccnn: Adaptive ccnn for density estimation and crowd counting," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 948–952.

[24] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[25] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification," in *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2017, pp. 1–7.

[26] G. Olmschenk, J. Chen, H. Tang, and Z. Zhu, "Dense crowd counting convolutional neural networks with minimal data using semi-supervised dual-goal generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition: Learning with Imperfect Data Workshop*, 2019.

[27] Z. Li, L. Zhang, Y. Fang, J. Wang, H. Xu, B. Yin, and H. Lu, "Deep people counting with faster r-cnn and correlation tracking," in *Proceedings of the International Conference on Internet Multimedia Computing and Service*, 2016, pp. 57–60.

[28] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3225–3234.

[29] Y. Hu, X. Jiang, X. Liu, B. Zhang, J. Han, X. Cao, and D. Doermann, "Nas-count: Counting-by-density with neural architecture search," in *European Conference on Computer Vision*. Springer, 2020, pp. 747–766.

[30] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.