Crowd Density Estimation using Imperfect Labels

Muhammad Asif Khan
Qatar Mobility Innovations Center (QMIC)
Qatar University
Doha, Qatar
mkhan@qu.edu.qa

Hamid Menouar
Qatar Mobility Innovations Center (QMIC)
Qatar University
Doha, Qatar
hamidm@qmic.com

Ridha Hamila
Electrical Engineering
Qatar University
Doha, Qatar
hamila@qu.edu.qa

2023 IEEE International Conference on Consumer Electronics (ICCE) | 978-1-6654-9130-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCE56470.2023.10043547

***Abstract*—Density estimation is one of the most widely used method for crowd counting in which a deep learning model learns from head annotated crowd images to estimate crowd density in unseen images. Typically, the learning performance of the model is highly impacted by the accuracy of the annotations and inaccurate annotations may lead to localization and counting errors during prediction. A significant amount of works exist on crowd counting using perfectly labelled datasets but none of these explore the impact of annotation errors on the model accuracy. In this paper, we investigate the impact of imperfect labels (both noisy and missing labels) on crowd counting accuracy. We propose a system that automatically generate imperfect labels using a deep learning model (called annotator) which are then used to train a new crowd counting model (target model). Our analysis on two crowd counting models and two benchmark datasets shows that the proposed scheme achieves accuracy closer to that of the model trained with perfect labels showing robustness of crowd models to annotation errors.**

***Index Terms*—annotations, Crowd counting, density estimation, federated learning, imperfect labels, noisy data**

I. Introduction

Crowd counting and density estimation is an important problem with many interesting applications. For instance, crowd count in public places such as political rallies, stadiums, and exhibition centers can be of great interest to authorities and event organizers for effective management and control actions. Vehicle counting is significant in transport planning, road intersection design, traffic signal control, and emergency vehicle preemption strategies. Similarly, wild life counting in forests can play a role to protect species of interest.

The diverse applications of crowd density estimation and counting has attracted the computer vision community. Over the past years, several state-of-the-art deep learning models using convolution neural networks (CNNs) have been proposed [1]–[10]. These models are trained with accurately labeled data (i.e., dot annotated images). However, the acquisition of dot annotated data can be expensive for dense crowds and large number of images. Moreover, the annotations may not be available beforehand in some scenarios due to security and privacy reasons.

In this paper, we propose an intriguing approach in which the network first generates density maps (imperfect and noisy) labels and then trains a target model on the generated labelled data. The motivation of the proposed approach is two-fold. First, dot-annotations is expensive and thus the automatic generation of ground truth density maps saves the time. Second, the perfect labels may not be available in some scenarios beforehand. For instance, deploying a number of drones (each running a lightweight model) for prediction can simultaneously fine-tune the model weights over time using federated learning [11]–[13]. However, the drones will need labels to predict count in the crowd images. Thus, one can use the proposed method to use a secondary deep model to predict the density map for the image which will serve as a ground truth to train the primary (target) model. The deep model will only be used for inference and when the node is participating in the federating learning round. Over the time, the lightweight target model will be fine-tuned and its accuracy will improve.

The contribution of the paper is as follows: We propose a novel method for training a lightweight crowd counting model by automatic label generation using a secondary deep crowd counting model (annotator). The auto-generated labels are typically imperfect and noisy. Whether such noisy and imperfect labels can be used to train another model? We trained two different models of different sizes and architectures on two benchmark datasets to evaluate the efficacy of the proposed method. The performance is investigated using standard metrics and some intriguing results are achieved.

II. Related Work

Density estimation using CNN for crowd counting was first proposed in [14]. The authors propose a simple single-column architecture with six layers. Several other works followed the approach and proposed different CNN architectures of various sizes typically to improve accuracy over benchmark datasets. The architectures used in these works include multi-column CNN [15]–[18], modular CNNs [3], [10], [19], encoder-decoder networks [6], [20], and transfer-learning based models [2], [7]–[9]. The initial small-sized single-column architectures (e.g. [14] generally achieve poor accuracy on images having large scale variations. Scale variations typically arise from the perspective distortions and different resolution images in the dataset. Thus, multi-column networks with filters of varying receptive fields in different column can be used to capture the scale variations. A multi-column convolution neural network

This publication was made possible by the PDRA award PDRA7-0606-21012 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

(MCNN) is proposed in [15]. MCNN is a three-column architecture with variable sizes filters ($9\times9, 7\times7, 5\times5$, and 3×3) in each column. The switching-CNN [17] address the scale variations differently. It uses three CNN networks (regressors) to process the input image using only one column based on the density in the image. The density is automatically estimated by another single column CNN (classifier or switch). One shortcoming of multi-column networks is their capability to adopt to the scale variations which is limited ny the number of columns i.e., more columns are needed when there are large variations in the images across the dataset. An alternative solution is to use modular networks which typically consist of a single column (sometimes multiple columns) architecture with scale-adaptive feature extraction modules. These models are mainly inspired from the Inception-like models [21]. Encoder-decoder models [6], [20] are also being used in in many research works. These models follow the popular UNet architecture [22], where the encoder part first learns and extracts features from the input image and then the decoder part generates prediction using the features passed by the encoder. Encoder-decoder models produce high quality density maps. Crowd counting in dense images and congested scenes can become more challenging. Thus, a significant amount of research contributions adopts transfer learning approach i.e., to use a pretrained image classification model e.g., VGG-16 [23], ResNet [24] or Inception [21] as a front-end (or backbone) for feature extraction and a shallow CNN network (back-end) to use the features for estimating the crowd density. These models are generally more accurate and faster to train but incur longer inference delays at prediction.

The aforementioned crowd counting models use accurately dot annotated localization maps as ground truth for the crowd images to train the model. All these works focus on investigating the accuracy improvement, but none of these explore the model performance over imperfectly labelled data.

III. Proposed Scheme

The proposed scheme is illustrated in Fig. 1. In first step, the system generates imperfect labels for the entire dataset using a deep network model (Annotator). The predictions using the annotator network serve as imperfect labels for training the target network. In the next step, the target network is trained on the noisy (imperfect labels). During the training process, the model learns from the imperfect labels by computing the loss function which is l_2 distance between the imperfect labels and the prediction of the target network. Then, to find the accuracy of the model, the mean absolute error (MAE) or l_1 distance between the predictions of the target network and the original ground truth (perfect) is calculated. In this way, the model learns from the imperfect data but its accuracy is tested on the actual ground truth.

We choose the CSRNet [8] model as the annotator network due to its good accuracy. However, CSRNet generates density maps of size 1/8 of the input image, whereas the target network used in our study generates density maps of size 1/4 of the input image. To solve this issue, we modified the original CSRNet [8] architecture. CSRNet uses the first 10 layers of VGG-16 [23] network. In VGG-16, pooling layers are uses to reduce the size of the predicted density map to half after layer-2 (1/2), layer-4 (1/4), layer-7 (1/8), and layer-10 (1/16). Using the first 10 layers (without pooling layer after layer-10), CSRNet generates output of size 1/8. In our modified version, We took only the first seven (7) layers (without the ppoling layer after layer-7) to generate output of size 1/4. Furthermore, the back-end network of CSRNet uses layers of sizes (512, 512, 512, 256, 128, 64), respectively. We also modified the back-end network to reduce layers sizes to (256, 256, 256, 128, 64, 64) to match the number of channels in the output of front-end and input of the back-end networks. We denote the modified CSRNet architecture as CSRNet_lite (due to relatively smaller size). It is shown in Fig. 2.

The CSRNet_lite has 3.9 Million parameters, almost four times less than CSRNet (16.2 Millions).

A. Model Training

The proposed scheme is evaluated over DroneRGBT dataset [25]. The DroneRGBT dataset has 1807 RGB and thermal image pairs in the train set. Each image has a spatial resolution of 512×640 pixels. We split the train set into a ratio of (70% : 30%) for training and validation. The dataset covers several scenes (e.g., campus, streets, public parks, car parking, stadiums, and plazas) and contains diverse crowd densities, illumination, and scale variations. The dataset provides head annotations of people.

In order to train CSRNet_lite model to generate imperfect labels, the dot annotations are first converted to ground truth (GT) density maps (perfect labels). To generate the GT density map the head positions (x_i) expressed as delta functions $\delta(x-x_i)$ are convolved with Gaussian kernels (G_σ).

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

where, N denotes the total number of dots (i.e., head positions) in the image. The value of $\sigma = 7$ is empirically determined by analyzing different values and visually seeing the overlapping between the blobs in the GT density maps. To prevent overfitting, we applied three different data augmentations techniques including random brightness, random contrast, and horizontal flipping. We use Adam algorithm [26] with an initial learning rate 0.0001 to optimize the loss function. To compute the loss, we use the pixel-wise euclidean distance between the target and predicted density maps as in Eq. 2.

$$L(\Theta) = \frac{1}{N}\sum_{1}^{N} ||D(X_i;\Theta) - D_i^{gt}||_2^2. \tag{2}$$

where N is the number of heads in the image, $D(X_i;\Theta)$ is the model prediction density map and D_i^{gt} is the GT density map. The predicted density maps are then used to generate another dataset with all the images of the original dataset whereas ground truth labels replaced by the predictions from CSRNet_lite. This noisy dataset is then used to

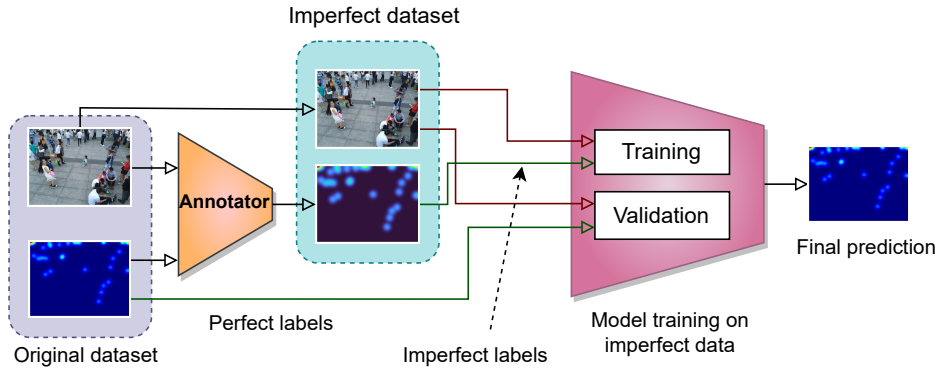

Fig. 1: Proposed method for automatic generation of ground truth and training a lightweight crowd counting model using imperfect (noisy annotation) dataset.

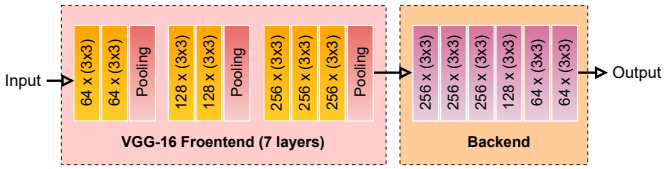

Fig. 2: CSRNet_lite: A modified lightweight version of CSRNet [8].

train a lightweight model. We use two lightweight models i.e., MCNN [15], and LCDnet (our own lightweight crowd density estimation model) to train from scratch using the noisy datasets. The same settings of optimizer and augmentation are used in the second stage as well. We use a single machine with two Nvidia RTX-8000 GPUs and PyTorch deep learning framework for training the model.

IV. Evaluation and Results

A. Evaluation Metrics

We evaluate the proposed method using the most widely used metrics i.e., mean absolute error (MAE), Grid Average Mean Error (GAME), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) calculated in Eq. 3, 4, 5, and 6, respectively. MAE and GAME measure the accuracy of the model by measuring the counting errors in GT and predicted density maps. The GAME metric is more sensitive to localization errors because it calculates MAE over patches of the predictions. The SSIM and PSNR measure the quality of the predicted density maps.

$$MAE = \frac{1}{N}\sum_{1}^{N}(e_n - \hat{g_n}). \tag{3}$$

where, N is the total number of images in the dataset, g_n is the ground truth (actual count) and $\hat{e_n}$ is the prediction (estimated count) in the n^{th} image.

$$GAME = \frac{1}{N}\sum_{n=1}^{N}(\sum_{l=1}^{4^L}|e_n^l - g_n^l|). \tag{4}$$

We set the value of $L = 4$, thus each density map is divided into a grid size of 4×4 creating 16 patches.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y C_2)}{(\mu_z^2\mu_y^2 + C_1)(\mu_z^2\mu_y^2 + C_2)}. \tag{5}$$

where $\mu_x, \mu_y, \sigma_x, \sigma_y$ represents the means and standard deviations of the actual and predicted density maps, respectively.

$$PSNR = 10log_{10}\left(\frac{Max(I^2)}{MSE}\right). \tag{6}$$

where $Max(I^2)$ the maximal in the image data (I). If it is an 8-bit unsigned integer data type, the $Max(I^2) = 255$.

B. Results

The first step in the proposed method is the generation of imperfect labels. Ideally, such imperfect labels should not have large errors. A sample label generated using the CSRNet_lite model for the given image and the ground truth density map is shown in Fig. 3.

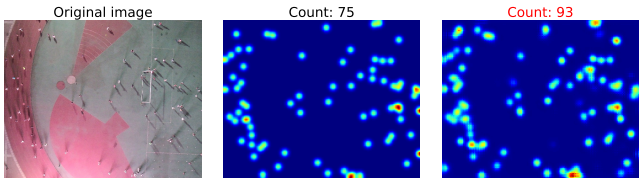

Fig. 3: The figure shows an image (left), its ground truth density map or perfect label (middle) generated from the dot annotation, and a predicted density map as imperfect label (right) generated using the CSRNet_lite model. The imperfect labels are used to train the lightweight model.

It can be observed that the noisy label is not perfect but is reasonably accurate to aid in training the target model. The performance of the two target models (i.e., MCNN and LCDnet) trained using the original target data (perfect data) and on noisy dataset is presented in Table I to show the learning performance using imperfect data.

TABLE I: Performance analysis of the proposed method using two lightweight models (MCNN and LCDnet) over two datasets (DroneRGBT, CARPK).

Model	Dataset	Labels	MAE	GAME	SSIM	PSNR
MCNN	DroneRGBT	Perfect	16.88	43.49	0.66	23.47
		Imperfect	17.86	46.92	0.64	23.07
MCNN	CARPK	Perfect	10.10	42.40	0.76	19.21
		Imperfect	10.86	43.92	0.64	18.03
LCDnet	DroneRGBT	Perfect	21.40	46.92	0.62	21.39
		Imperfect	23.82	47.91	0.60	20.07
LCDnet	CARPK	Perfect	13.10	46.13	0.69	20.14
		Imperfect	13.75	48.26	0.66	20.07

One can observe that both networks (MCNN and LCDnet) when trained over imperfect labels show very good accuracy on test (unseen) data. There is very small or sometimes negligible difference in the MAE and GAME metrics for both models when trained on the two datasets using perfect and imperfect datasets, respectively. Fig. 4 show sample prediction over DroneRGBT [25] dataset using the proposed scheme (showing both predictions over perfect and noisy datasets used in training). On most images, the predicted count was exactly same for both predictions (using models trained over perfect and noisy data). This means, that even imperfect data with missing or noisy annotations (due to prediction errors) can be used by the model to learn features which are even not learnt by the annotator network.

C. Ablation Study

Above we discussed the model learning performance over noisy or imperfect training data (generated by another model). The analysis shows that the model can learn features (objects) which are even not annotated in the training data from other annotations and similar features. To further investigate the learning process, we train and evaluate the two models using missing annotations instead of imperfect data (noisy labels). To implement this, we randomly delete a fixed portion (30%) of annotations from each image to generate imperfect labels (see Fig. 5).

The two models MCNN and LCDnet are then trained over this dataset with missing labels. The model performance is compared with the same model trained with perfect data. As anticipated, the model could learn unlabelled objects of interest. A comparison over CARPK [27] dataset is shown in Table II.

The results in Tab;e II strengthen the idea that having a good model and efficient training strategy can overcome the annotation errors (missing and noisy annotations) in the training data.

TABLE II: Ablation study using imperfect data (missing labels).

Model	Dataset	Annotation	MAE	GAME
MCNN	CARPK	Perfect	10.1	43.4
		Missing labels	14.4	53.6
LCDnet	CARPK	Perfect	13.7	48.2
		Missing labels	18.5	55.3

V. Conclusion

The performance of any deep learning model is hugely correlated with the quality of the data. More specifically, accurate labels improve the learning process whereas noisy or imperfect labelled data can lead to inaccurate predictions. In crowd density estimation, the ground truth labels contains dot annotations of all heads (or object centers) in the image. Dot annotation is expensive and hence we propose to automatically generate imperfect labels by using a deep CNN model and then investigate the learning performance of a shallow model on the imperfect data. The results report that even the imperfect data can be efficiently used to train another model from scratch. The proposed scheme can be efficiently used in fine-tuning crowd counting models running over distributed devices (such as drones) in federated learning environment without human-aided labelling. As a future work, we plan to apply the proposed method in other computer vision based applications such as object detection and anomaly detection.

References

[1] M. A. Khan, H. Menouar, and R. Hamila, "Revisiting crowd counting: State-of-the-art, trends, and future perspectives," *ArXiv*, vol. abs/2209.07271, 2022.

[2] H. Tang, Y. Wang, and L.-P. Chau, "Tafnet: A three-stream adaptive fusion network for rgb-t crowd counting," *ArXiv*, vol. abs/2202.08517, 2022.

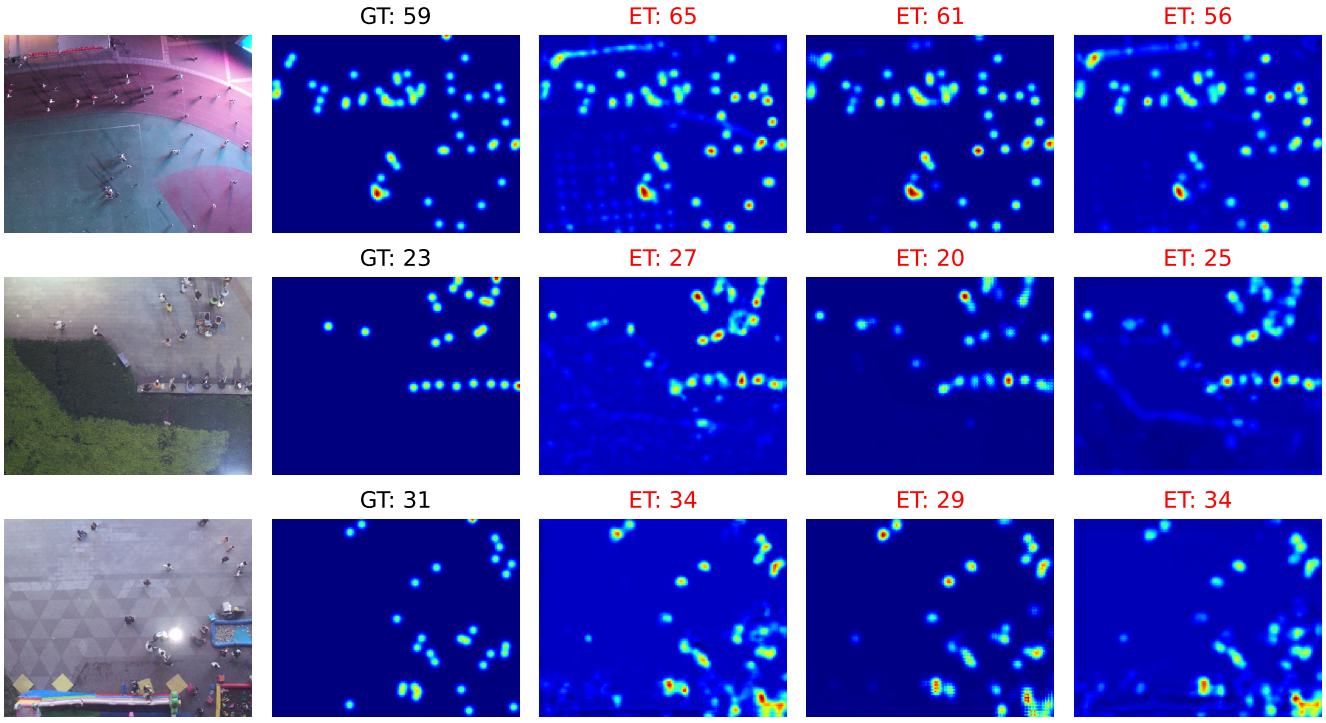

Fig. 4: Sample predictions on DroneRGBT dataset [25]: Column 1 shows same images from the dataset. Column 2 shows ground truth (perfect). Column 3 shows predictions of the model trained on the perfect ground truth. Column 4 shows imperfect labels (generated using CSRNet_2 network. Column 5 shows predictions using the model trained on imperfect data. GT means ground truth (actual count), whereas ET denotes estimated truth (predicted count).

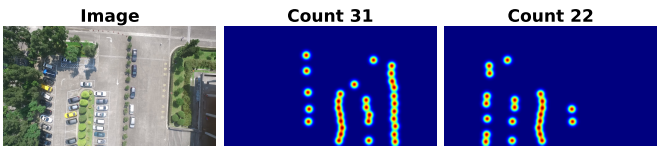

Fig. 5: A sample ground truth (GT) density map with missing (30%) data.

[3] Q. Wang and T. Breckon, "Crowd counting via segmentation guided attention networks and curriculum loss," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[4] Q. Song, C. Wang, Y. Wang, Y. Tai, C. Wang, J. Li, J. Wu, and J. Ma, "To choose or to fuse? scale selection for crowd counting," in *AAAI*, 2021.

[5] Z. Chen, J. Cheng, Y. Yuan, D. Liao, Y. Li, and J. Lv, "Deep density-aware count regressor," in *ECAI*, 2020.

[6] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. S. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6126–6135, 2019.

[7] W. Liu, M. Salzmann, and P. V. Fua, "Context-aware crowd counting," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5094–5103, 2019.

[8] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1091–1100, 2018.

[9] S. Aich and I. Stavness, "Global sum pooling: A generalization trick for object counting with small datasets of large images," *arXiv preprint arXiv:1805.11123*, 2018.

[10] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *ECCV*, 2018.

[11] X. Liu, Y. Deng, and T. Mahmoodi, "A novel hybrid split and federated learning architecture in wireless uav networks," in *ICC 2022 - IEEE International Conference on Communications*, pp. 1–6, 2022.

[12] M. O. Osifeko, G. P. Hancke, and A. M. Abu-Mahfouz, "Surveilnet: A lightweight anomaly detection system for cooperative iot surveillance networks," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25293–25306, 2021.

[13] D. Unal, M. Hammoudeh, M. A. Khan, A. Abuarqoub, G. Epiphaniou, and R. Hamila, "Integration of federated machine learning and blockchain for the provision of secure big data analytics for internet of things," *Computers & Security*, vol. 109, p. 102393, 2021.

[14] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–841, 2015.

[15] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597, 2016.

[16] L. Boominathan, S. S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

[17] D. Sam, S. Surya, and R. Babu, "Switching convolutional neural network for crowd counting," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 4031–4039, IEEE Computer Society, jul 2017.

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

[19] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 465–469, 2017.

[20] C. Gao, P. Wang, and Y. Gao, "Mobilecount: An efficient encoder-decoder framework for real-time crowd counting," in *Pattern Recognition and Computer Vision: Second Chinese Conference, PRCV 2019, Xi'an, China, November 8–11, 2019, Proceedings, Part II*, p. 582–595, Springer-Verlag, 2019.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv*, vol. abs/1505.04597, 2015.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[25] T. Peng, Q. Li, and P. Zhu, "Rgb-t crowd counting from drone: A benchmark and mmccn network," in *Computer Vision – ACCV 2020: 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 – December 4, 2020, Revised Selected Papers, Part VI*, (Berlin, Heidelberg), p. 497–513, Springer-Verlag, 2020.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[27] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4165–4173, 2017.