Learning Weight of Losses in Multi-Scale Crowd Counting

1st Derya Uysal
Department of Computer Engineer
Istanbul Technical University
Istanbul, Turkey
deryaguler1995@gmail.com

2nd Uluğ Bayazıt
Department of Computer Engineer
Istanbul Technical University
Istanbul, Turkey
ulug.bayazit@itu.edu.tr

2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA) | 979-8-3503-3890-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/INISTA59065.2023.10310484

***Abstract*—In this work, we improve the state-of-the-art in crowd counting by further developing a recently proposed multi-scale, and multi-task crowd counting approach. While most of the studies treat density-based architectures, this study proposed a point-based method for crowd analysis. We propose automatic, and optimal weight assignment to constituents of the loss function. This approach, which is applied to each patch, ensures that the weight parameters are updated in each epoch, and added to the optimizer with model parameters rather than remaining constant. For validation of our proposed approach, we use three popular crowd counting datasets, ShanghaiTech A, ShanghaiTech B, and UCF_CC_50. The performance of our approach exceeds the performances of the other studies on the ShanghaiTech dataset, and is highly competitive with the performances of the other studies on the UCF CC 50 dataset.**

Keywords—Crowd Counting, Multi Scale, Automatic Weighted Loss, Point Supervision

I. Introduction

The goal of crowd counting, is to determine how many individuals are visible in a particular image or video. Falling within the interest domain of video surveillance, density estimation [1, 2, 3], congestion estimation, segmentation, crowd management, public safety, scene understanding, and anomaly detection, the problem of accurately predicting crowd counts has become an important application area of computer vision, and received a lot of attention in recent years. State-of-the-art counting techniques have also been used in other fields, such as automatic crowd management [1, 2] counting cells or germs from microscopic photos [12], crop yield estimation [4, 5], estimating the number of animals [6], calculating the number of cars during traffic jamps[7], traffic monitoring [8], etc. For instance, in the COVID-19 epidemic, it is important to keep an eye on social distance, and prevent the disease's spread [9]. All things considered, this problem is difficult due to the factors such as the large-scales, and crowd images with perspective, and bluring in the data. As the same amount of people might have significantly different crowd distributions, it is not sufficient to just count the number of crowd population in such real-world applications.

For some images, such as blurry or very crowded images with different camera perspectives, it is very difficult to do crowd analysis without CNN [14, 22, 16, 18, 17, 12, 11, 23, 24]. Therefore, most recent studies use CNN architecture. Modern techniques for analyzing crowded settings range from straightforward crowd counts to density map presentations that tag head locations, and convert with a Gaussian kernel [2], but this approach requires large kernel sizes, and these studies just calculate the number of people in a scale without localization. Multi-scale structures form the foundation for the majority of recent CNN based crowd counting studies have enabled high performance. On the other hand, traditional methods, such as detection, and segmentation, have been unsuccessful due to the indistinctness of the shapes of the people in the image.

In this paper, we propose the use of a weighted loss function with log-likelihood of Gaussian distribution [10] that effectively merges all the patch losses in the recently proposed multiscale crowd counting approach of [11]. The loss weights are updated together with the other parameters to reach optimal values of the model in every learning epoch. In this study, we used points that indicate head location instead of a density map. In this way, we can find the position of the people in the images not just count of them. Additionally, we use a comprehensive dataset (ShanghaiTech A, ShanghaiTech B, and UCF_CC_50) to compare our method against other state-of-the-art methods in the literature. Our results show performance improvement over the approach of [11].

II. Related Work

The literature contains many proposed algorithms for crowd counting. Some of the primary categories for crowd counting methods are density-based, and point-based. Some of these work based on CNNs, detection, and regression. Studies conducted using a CNN-based approach have shown greater success, and faster results compared to traditional methods. The first studies in this field were conducted using a density-based approach, and maps of the images were created to calculate loss over them. On the other hand, point-based studies have calculated loss through people counting. This section will discuss the studies conducted using these methods.

Most of the first researchers focus on the detection-based method to detect people, and count them. Some studies are designed to classify objects, and detect the human body. With some experiments, some of the just focused on head location, and detected them. However, these studies are so poor, and insufficient to estimate the crowd in a given area.

Regression-based studies were generally used in the past, and they were not just used for crowd analysis. Especially some object counting areas like product, anlimal, etc. were chosen this way [5, 12, 13].

Most of the state-of-the-art methods use a density map to obtain objects in a space. Gaussian filters are typically applied to the head position area to create a density map. Predictions are then obtained as a probability based on the generated density map. The first was used in [25], and they found a solution with linear mapping like a density map. After this solution, many studies in that area used this technique to solve the counting problem. CSRNet [12]

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

utilized dilated convolutions to effectively expand its receptive field, capturing contextual information, and obtaining strong crowd image representations. Switch network [13] used independent CNN regressors, and these regressors have different receptive fields to count people on scales. In CrowdNet [3], a fully convolutional network was used to predict a density map with shallow, and deep branches. Both of these methods were used to detect high, and low level information. MSCNN [14] network obtained scale-relevant features of varying sizes. MCNN [2] used three branches of CNN to predict objects with different receptive fields. In [1], a data-driven method into patches was developed to work on unseen taget scene. In [4], multi-kernel pooling, and stacked pooling were proposed to capture scale-adaptive features in a scene. Sanet [15] extracts the features with scale egregation modules, and in addition, they use a novel loss function with Euclidean loss. CP-CNN [16] includes four modules, such as global context, local context, density map, and fusion CNN, to estimate contextual information. IC-CNN [17] generates a model with two branches for high resolution. One of them is for low resolution, and the other one predicts features with the first branch output. IGCNN [18] used a density regressor with growing CNN to the wide viability seen in scenes. On the scenes in [19], a proposed classification method is shown. This architecture separates closed sets to protect against errors on unseen scenes. The regions divide the sub-region until the count falls below the threshold. After the counting of the region , the architecture combines all of the sub-regions, and calculates the total of the people count.

Some studies use a point-based approach, and the biggest advantage is localization without any density map or bounding box. Especially in this paper, we use a base study to generate a new approach for reducing the loss. In [14] proposed a multiscale, and multitask point-level approach that uses a pretrained model, and does localization.

Particularly CNN-based method was proposed in this field to estimate recognition, and classification because of better performance, and success. Multi-scale representations was used in CrowdNet [3] to be more efficient, and it proposed two detector mechanism as face/body, and blob to get high, and low level information on the inputs. In [20] produced single column fully convolutional network model, and setting agumentation method to minimize loss during training finally all of these was generated with multi-scale. Classification methos can be used in [21], thera are some crowd area, and architecture classfy scenes also estimate the amount of people in the image. It is clear to see CNN based methohs have more power, and efficient according to other tradional methods. An adversarial loss is immediately used to reduce the solution, attenuating the hazy impacts of density map estimation. In [22] A U-net structured creation network is constructed to build density map from input patch. On the other hand, an unique scale consistency regularizer has been developed that mandates that the total of crowd counts come from local patches. P2PNet [23] skips unnecessary stages, and estimates a collection of point suggestions to describe heads in a picture in a way that is compatible with the findings of human annotation. The intrinsic relationship between the scales, and the feature levels is instantly learned by SASNet [24].

III. Proposed Method

In this section, the normalization operations for the dataset, and the weighting mechanism added to the architecture are described. The details of two different methods used to creating construct this method are mentioned. We also explain the multi-task loss function based on a set of fundamental principles.

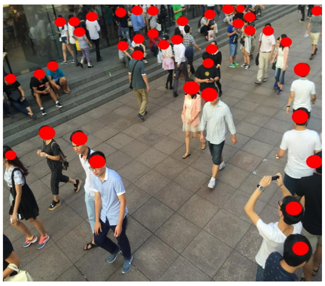

Fig. 1. A sample image of the dataset, and the head locations marked by using the *(x,y)* coordinate information

A. *The Data and its Preprocessing*

In a point-based approach similar to the multiscale crowd counting method of [11], we employ the coordinates of the people in an image to determine their count. To calculate the loss, there needs to be ground truth head locations at every scale. Hence all coordinates for different scales are normalized before being used [11]. The datasets contain *(x,y)* coordinates of people, and the total number of people in each image. A sample image is shown in Figure 1. The first stage of the architecture applies preprocessing to the input image data. Pixels corresponding to *(x,y)* coordinate labels are labelled with value 1, and the other pixels are labelled with value 0. We derive four different sets of coordinates for a point for four different scales, by using normalization. First, x-coordinates are divided by width, and y-coordinates by height of the patch at full scale. Then, for a patch at a lower scale, these values are multiplied by the height, and width of this patch as suggested by Eq. 1 and Eq. 2 (x_i, shows coordinate in a scene, and *j* shows scale parameter like 1/2, 1/4, 1/8, 1/16, *w* and *h* show the weight and height of the image).

$$x_i^j = \left(\frac{x_i}{w}\right) * (w * j) \tag{1}$$

$$y_i^j = \left(\frac{y_i}{h}\right) * (h * j) \tag{2}$$

B. *Method*

The current work is based on the auxiliary tasks in multitask learning study [10] and the multitask and multiscale crowd counting study [11].

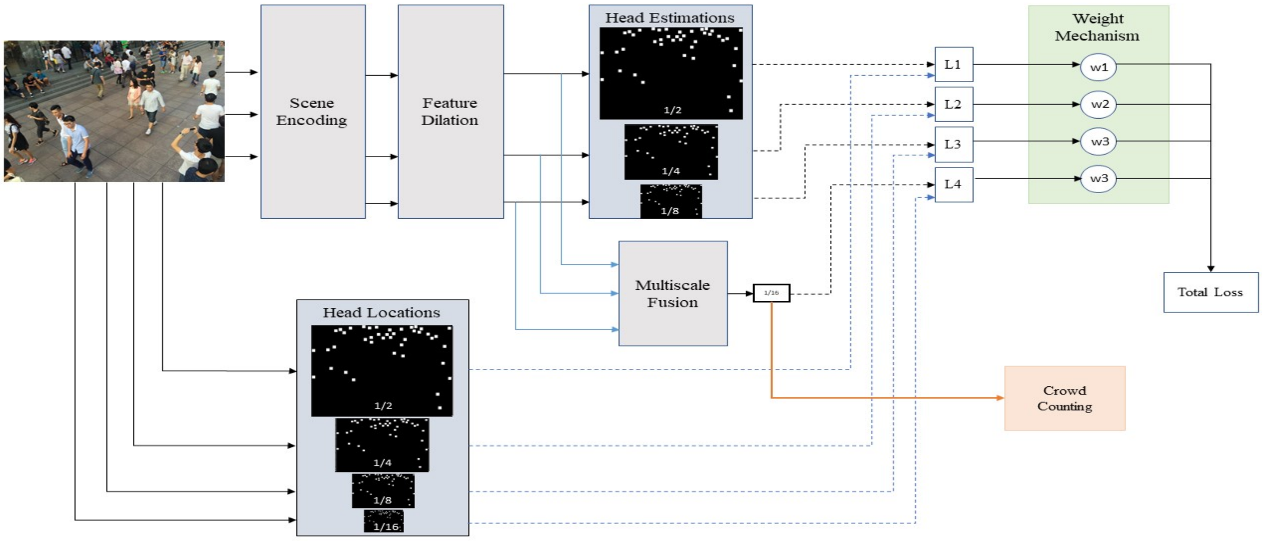

Fig 2. The multiscale crowd counting system architecture of [11], and the proposed automatic weight mechanism (shown in green box)

The architecture of the multitask and multiscale crowd counting study was adopted in this study and the method of the auxiliary tasks in multitask learning [10] was used for optimization of the loss function. The automatic weighting approach of [10] updates all branch (scale) losses in every step in order to decrease the overall loss, and increase performance of the system. In this study, they developed the loglikelihood method from the method used in [27].

In the scene geometry, and semantic study [27] was used to determine the weights of the loss functions of multi-tasks such as classification, and regression. Homoscedastic uncertainty is a weighting system that can learn to balance the losses of many problems such as classification, and regression, and shows how to create multi-loss functions [27]. They develop the probabilistic model and establish a multi-task loss function. They describe likelihood for regression tasks as a Gaussian by using average determined by the model output and a noise parameter. The log likelihood in Eq. 3 is maximized in relation to the model parameters and the observation noise parameter. The $f^{W}(x)$ function on the left side of this equation shows the output y obtained with x as the input of the neural network and where W are the weights and σ is the observation noise scalar. This model is termed the probabilistic model.

$$\log p(y|f^{W}(x)) = -\frac{1}{2\sigma^2}\left\|y - f^{W}(x)\right\|^2 - \log\sigma \qquad (3)$$

In the multiscale crowd counting approach of [11], there are four different branches. Each branch includes some specific information like density at a particular scale where branches of larger scales contribute more detailed information.

In Figure 2, the green part is our proposed contribution to the system of [11], which consists of components such as scene encoding, feature dilation, multiscale fusion, head estimation, and head location. In scene encoding, the top layers of the pretrained model, VGG-16 [26] have been truncated at different heights to obtain the different scales of S1 = 1/2, S2 = 1/4, and S3 = 1/8 of the input scene as shown in Figure 3. Thereby, the multiscale crowd counting [11] method uses the three patches at these different scales to get more detailed information.

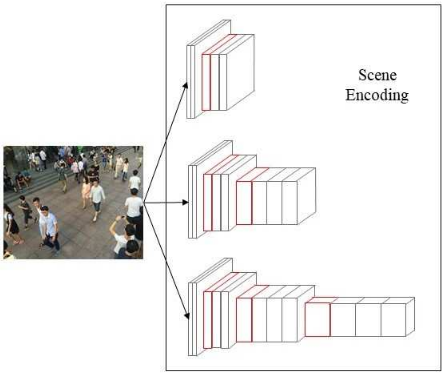

Fig. 3. Network structure of Scene Encoding (Max pooling, and Conv layers with Leaky ReLU)

The feature dilation part is used with a dilated convolution layer to extract information from the scene for every pixel from a wider region as shown in Figure 4. Different numbers of layers are applied to different branches to ensure consistency between different branches, and then multiscale fusion is applied to get the final branch output by concatenating the outputs of the three branches at different scales. In multiscale fusion, a density map is formed using feature maps at different scales to make counting estimates. The last part of the model uses a one-channel layer to extract the location on an image for every branch.

The final loss in [11] is a linear combination of the losses for different branch outputs as suggested by Eq. 4. In [11], the weights had fixed values like 0.1, 0.2, 0.3, and 0.1 for the different branches. Iterative, application of such manual configuration to determine the optimal weights in a trial, and error strategy is conceivably quite time consuming and expensive.

$$L_{comb} = \sum_{i=1}^{4} w_i L_i \qquad (4)$$

Auxiliary tasks in multitask learning study [10] proposes a multitask architecture for scene understanding problems. All of the losses for the different tasks are combined with the weight parameters to get the multi-task loss. The weight parameters are configured as parameters that can be learned in the network rather than manually tuned.

By using, the log likelihood of Gaussian distribution the loss function may be written as a sum of regularized, weighted branch losses as given by. We calculate the branch losses with MSE Eq. 6 (L_i: ith branch loss, w_i: ith branch weight, mean of square differences of the pixel values labelled with 0 and 1 according to presence and absence of a person head at the pixel locations).

$$l_{total} = \sum_{i=1}^{4} \left(\frac{1}{2} * \frac{L_i}{w_i^2} + \ln(1 + w_i^2) \right) \tag{5}$$

In Eq. 5., the weight term w^2 assumes the role of the observation noise variance σ^2 in Gaussian distribution. Furthermore, the $log(w^2)$ term is viewed as a regularization term which is changed to $ln(1+w^2)$ in order to ensure non-negative contributions to the total loss.

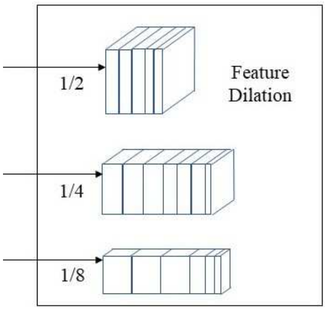

Fig. 4. Network structure of Feature Dilation (Dilated Conv layers)

In order to improve the manual weight tuning of [11] for crowd counting by applying the approach of [10] the weight parameters (noise standard deviations) are learned together with the network parameters.

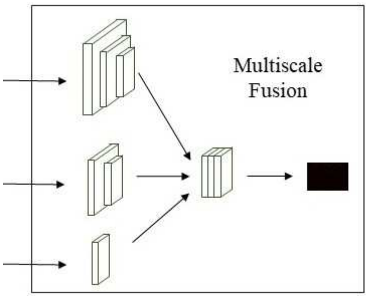

Fig. 5. Network structure of Multiscale Fusion (Conv layers with stride, and Leaky ReLU)

The branch loss weight parameters are supplied to the optimizer function along with the model parameters to get the minimum value for the overall loss as in [10]. As opposed to learning each task separately, this training technique balances the weights appropriately, and yields better performance.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y})^2 \tag{6}$$

Since the dimension of the search space for the weight combinations is low, the overall loss function is insensitive to the choice for weight initialization, and an initial weight combination appropriate for the data of interest can be specified. In this study, the starting points were selected as 0.1, 0.2, 0.3 and 0.1 based on the weight parameters selected in [11].

- After calculating the branch losses with MSE Eq. 6, by using the initial weight values, we calculate loglikelihood Eq. 5, and weight parameters after each prediction.
- We then include the weight parameters in the optimization function, They are updated together with the model parameters at each step.
- Then we calculate the linear combination Eq. 4 using the weight parameters and loss values to compare with base method.

IV. Experiments And Results

In this study, we have used three datasets that are most popular in crowd counting: ShanghaiTech A, ShanghaiTech B, and UCF_CC_50. ShanghaiTech A has 482 images with train, and test data, while ShanghaiTech B has 716 images with two data parts. ShanghaiTech A is a dataset collected from internet images, and ShanghaiTech B includes images from street cameras. Additionally, UCF_CC_50 has 50 images.

The train, and test parts of the data were combined in our experimental work, and the five fold cross-validation procedure was used. This way, instead of reporting the performance on specific test images, the reported performance figures are averages over the entire dataset.

The Adam optimizer was used with the automatic weight parameters, and the learning rate was 1E-5. For obtaining the results on the datasets ShanghaiTect A, and ShanghaiTect B, the model was trained for 90-epochs, whereas it was trained for 70-epochs for obtaining the results on UCF_CC_50. These steps were chosen because of overfit when trained with more epochs. We used the last branch to calculate crowd counting as part of the study. In The MSE, and MAE losses expressed in Eq. 6 and Eq. 7 (y: ground truth value, $\hat{y}$: predicted value, mean of absolute difference of the pixel values labelled with 0 and 1 according to presence and absence of a person head at the pixel locations) are used to compare the proposed method against the state-of-the-art methods.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}| \tag{7}$$

The results obtained with the compared approaches on the ShanghaiTech A, and ShanghaiTech B datasets, and the UCF_CC_50 datasets are shown in Table 1, and Table 2, respectively. The proposed method helps to reduce the loss, and gets better performance compared to the other studies.

With this way, ShanghaiTech is better than multiscale crowd counting, and UCF_CC_50 is better than other recent technologies based on density-based, and CNN-based approaches.

TABLE I. MAE AND MSE RESULTS ON THE SHANGHAITECH A AND SHANGHAITECH B

Dataset	ShanghaiTech A		ShanghaiTech B	
	MAE	MSE	MAE	MSE
Zhang et al. [1]	181.8	277.7	32	49.8
MCNN [2]	110.2	173.2	26.4	41.3
Cascaded-MTL [21]	101.3	152.4	20.0	31.1
Swithching CNN [13]	90.4	135.0	21.6	33.4
MSCNN [14]	83.8	127.4	17.7	30.2
ACSCP [22]	75.7	102.7	17.2	27.4
CPCNN [16]	73.6	106.4	20.1	30.1
IGCNN [18]	72.5	118.2	13.6	21.1
ICCNN [17]	68.5	116.2	10.7	16.0
CSRNet [12]	68.2	115.0	10.6	16
Multiscale CC [11]	71.4	110.7	9.6	15.0
Ours	**65.2**	**97.22**	**9.25**	**14.5**

TABLE II. MAE AND MSE RESULTS ON THE UCF_CC_50

Dataset	UCF_CC_50	
	MAE	MSE
Zhang et al. [1]	467.0	498.0
MCNN [2]	377.6	509.1
Cascaded-MTL [21]	322.8	341.4
Swithching CNN [13]	318.1	439.2
MSCNN [14]	363.7	468.4
ACSCP [22]	291.0	404.6
CPCNN [16]	295.8	320.9
IGCNN [18]	291.4	349.4
ICCNN [17]	260.9	365.5
CSRNet [12]	266.0	397.0
SANET [15]	258.4	334.9
SDCNet [19]	204.2	301.3
P2PNet [23]	172.72	256.18
To choose or fuse [24]	161.4	234.46
Ours	**153.1**	**208.02**

We have also tested the validity of the employed architecture by conducting certain ablation studies. Firstly, we have used only the branch with the largest scale (1/2) in the multiscale fusion section, by removing the concatenate operation, but we used multiscale fusion layers, so the size is 1/16 scale. This yielded a high error rate (MAE: 164, MSE: 247 for ShanghaiTech A). A second study was conducted in the scene encoding part by cancelling the downscaling for all branches. This resulted in a scale of 1/8 which did not bring down the error rates either (MAE: 118, MSE: 183 for ShanghaiTech A). For this reason, we conclude that each of the four different branches contributes to the performance.

In Fig. 4, the results of some examples for ShanghaiTech part A and part B are given. The white and red dots give the total ground truth points. The red dots in the picture are the points that the model cannot predict, and the green dots are the points that the model predicts. In general, it has been observed that the proposed method has difficulty estimating distant head points. It was seen that the estimated green dots were estimated from places close to the white dots. There are as many green dots as each white dot, and the model could not predict the red dots. Likewise, the results of the UCF_CC_50 dataset can be seen in Fig. 5. Since the pictures here contain very crowded or blurry images, the error rate is higher. If we look at the images on the right, the mistakes made are common throughout the picture, since the images are usually very crowded and seem to be taken from above. Since there is no perspective in the picture, we cannot comment in terms of distance.

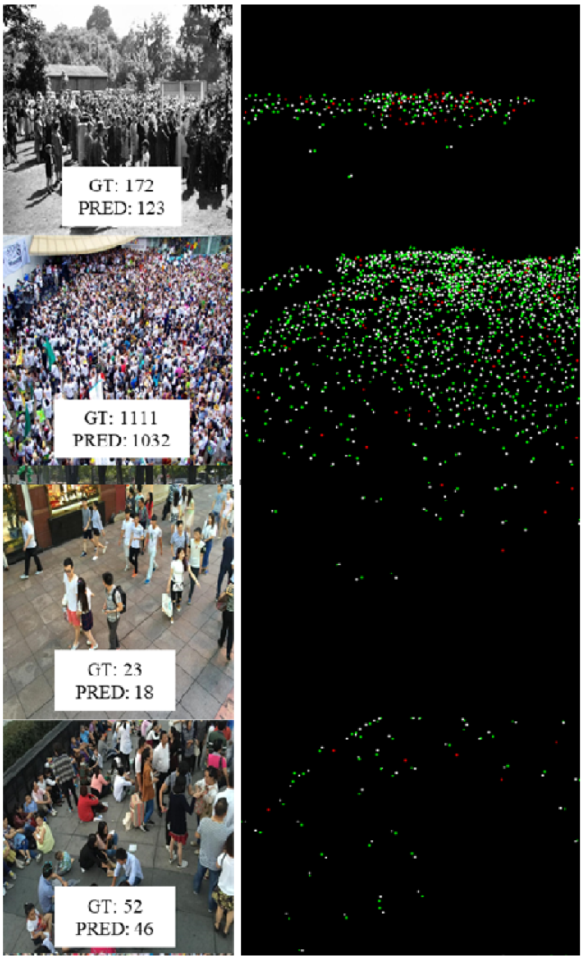

Fig. 4. Results of some samples for ShanghaiTech part A and part B.

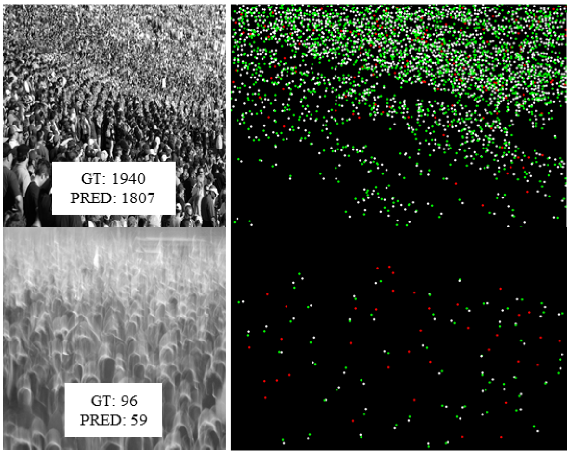

Fig. 5. Results of some samples for UCF_CC_50.

V. Conclusion

In this paper, we use an automatic weight loss approach to calculate total loss during the train while optimizing these parameters at each step with the model parameters. Therefore, the results are better than the previous work. Additionally, we have used cross validation to increase the generalization ability of the model for unseen data, and we have employed the UCF_CC_50 dataset to compare our proposed method to other state-of-the-art methods. Our experiments show that results on ShanghaiTech Part A, and Part B are better than the previous multiscale crowd counting study. On the other hand, UCF_CC_50 results are better than the recent methods such as P2PNet, SANet, etc.

For future work, we can add some mechanisms to improve the performance of the model in the architecture like breaking up large images into smaller ones. At the same time, images with high error rates can be viewed during training and extra preprocessing (such as sharpening blurry images, breaking images that are too crowded into smaller pieces) can be performed on these images. Additionally, this mechanism can be used with the density map approach for crowd counting or other object counting problems.

References

[1] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In Proceedings of the IEEE CVPR, pages 833–841, 2015.

[2] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Singleimage crowd counting via multi-column convolutional neural network. In Proceedings of the IEEE CVPR, pages 589– 597, 2016.

[3] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In Proceedings of the 2016 ACM on Multimedia Conference, pages 640–644. ACM, 2016.

[4] Jose A. Fernandez-Gallego, Shawn C. Kefauver, Nieves Aparicio Gutierrez, Mar ´ ´ıa Teresa Nieto-Taladriz, and Jose Luis Araus. Wheat ear counting in-field conditions: ´ high throughput and low-cost approach using RGB images. Plant Methods, 14(1):22–33, 2018.

[5] Hao Lu, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen. TasselNet: counting maize tassels in the wild via local counts regression network. In The Plant Methods, 13(1):79– 95, 2017.

[6] Marco Willi, Ross T. Pitman, Anabelle W. Cardoso, Christina Locke, Alexandra Swanson, Amy Boyer, Marten Veldthuis, Lucy Fortson. Identifying animal species in camera trap images using deep learning and citizen science. In The Methods in Ecology and Evolution, 10(1):80-91, 2018.

[7] Henrik Bette, Lars Habel, Thorsten Emig, Michael Schreckenberg. Mechanisms of jamming in the Nagel-Schreckenberg model for traffic flow. In The Physical Review E 95(1), 2017.

[8] Daniel Onoro-Rubio and Roberto J. Lopez-Sastre. Towards perspective-free object counting with deep learning. In The European Conference on Computer Vision (ECCV), pages 615–629, 2016.

[9] Mohammad Al-Sa'd, Serkan Kiranyaz, Iftikhar Ahmad, Christian Sundell, Matti Vakkuri and Moncef Gabbouj. A Social Distance Estimation and Crowd Monitoring System for Surveillance Cameras. In The Computer Visions and Pattern Recognition, 22(2), 418, 2022.

[10] Lukas Liebel and Marco Korner. Auxiliary tasks in multi-task learning. arXiv preprint arXiv:1805.06334, 2018.

[11] Mohsen Zand, Haleh Damirchi, Andrew Farley, Mahdiyar Molahasani, Michael Greenspan, Ali Etemad. Multiscale Crowd Counting and Localization by Multitask Point Supervision. 2022.

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

[13] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 4031–4039.

[14] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multiscale convolutional neural networks for crowd counting," in IEEE International Conference on Image Processing, 2017, pp. 465–469.

[15] 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), pages 734–750, 2018.

[16] Vishwanath A. Sindagi and Vishal M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In The IEEE International Conference on Computer Vision (ICCV), pages 1861–1870, 2017.

[17] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In The European Conference on Computer Vision (ECCV), pages 270–285, 2018.

[18] Deepak Babu Sam, Neeraj N. Sajjan, R. Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3618–3626, 2018.

[19] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Count ing objects by spatial divide-and-conquer. In IEEE International Conference on Computer Vision, 2019.

[20] Mark Marsden, Kevin McGuiness, Suzanne Little, and Noel E O'Connor. Fully convolutional crowd counting on highly congested scenes. arXiv preprint arXiv:1612.00220, 2016.

[21] 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.

[22] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5245–5254, 2018.

[23] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Yang Wu. Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework. ICCV, pages 3365-3374, 2021.

[24] Qingyu Song, Changan Wang1, Yabiao Wang1, Ying Tai1, Chengjie Wang, Jilin Li, Jian Wu2, Jiayi Ma. To Choose or to Fuse? Scale Selection for Crowd Counting. The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), pages 2576-2583, 2021.

[25] V. Lempitsky and A. Zisserman. Learning to count objects in images. In NIPS, pages 1324–1332, 2010.

[26] K. Simonyan and A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR, 2015.

[27] A. Kendall, Y.Gal, and R. Cipolla. Multi-task learning using uncertainty to weight losses for scene geometry and semantics. In Conference on Computer Vision and Pattern Recognition, 2018.