Comparison YOLOv5 Family for Human Crowd Detection

https://doi.org/10.3991/ijoe.v19i04.39095

Mohammed Abdul Jaleel Maktoof[1(✉)], Israa Tahseen Ali Al_attar[1], Ibraheem Nadher Ibraheem[2]
[1]Computer Science Department, University of Technology, Baghdad, Iraq
[2]Faculty of Basic Education, Mustansiriyah University, Baghdad, Iraq
abdeljaleelmohammed@gmail.com

Abstract—Recent years have seen widespread application of crowd counting and detection technology in areas as varied as urban preventing crime, station crowd statistics, and people flow studies. However, getting accurate placements and improving audience counting performance in dense scenes still has challenges, and it pays to devote a lot of effort to it. In this paper, crowd counting models are proposed based on the YOLOv5 algorithm, and four YOLOv5 models (YOLOv5l, YOLOv5m, YOLOv5s, YOLOv5x) were built for the purpose of comparing the models and increasing the accuracy of crowd identification as each model contains certain characteristics such as Filter sizes. Each model was trained on a human dataset (indoor and outdoor) for the purpose of comparing the results of each model and showing which model reaches higher accuracy in detecting people. Through this study and practical experiments conducted on each model, it was found that the best model is YOLOv5x, and YOLOv5l, where the accuracy of detecting humans reached more than 96%, while YOLOv5s reached more than 92%, and YOLOv5m reached the lowest accuracy, which is 91%.

Keywords—crowd detection, crowd counting, deep learning, YOLOv5

1 Introduction

In recent years, computer vision has been used to count dense crowds. It can be used to estimate the size of political rallies, civil unrest, social and sporting events, etc., as well as to count the number of participants. Crowd counting methods also offer significant potential to perform similar tasks in other fields, such as traffic congestion estimation [1], cell and bacterial count from microscopic imaging [2], and animal crowd estimations for ecology surveys [3], to name a few. Perspective effects and occlusions between each other can make them look very different in shape, size, and appearance in the images. During the last ten years, a number of algorithms [4] for counting crowds have been proposed in the literature. In recent times, methods for counting crowds that use Deep Learning algorithms have come a long way [5–8]. The best approaches rely on density map estimating, which predicts a density map for the input data and sums it. The best approaches rely on density map estimating, which predicts a density map for

the input data and sums it.The current gold standard for making use of these annotations is to create a "ground truth" by transforming the point annotations for each training image, and then regress each pixel in the density map to train a YOLO model [9]. The performance of a YOLO model is dependent on the quality of the "ground-truth" density maps obtained under these conditions [10].

2 Related works

The most common related works in crowd detection and counting in literatures are:

In 2022 [11], offer a deep neural network architecture for multi-view crowd counting, which integrates information from numerous camera perspectives to forecast a scene-level density map in three – dimensional. Take into consideration the following three variations of the fusion structure: the late fusion design fuses the camera-view density map; the naive early fusion design fuses the camera-view feature maps; and the multi-view multi-scale early fusion design makes sure that features aligned to the identical ground-plane point have consistent scale items. In addition, feature rotation alignment consistency is guaranteed by a rotation selection module. All three of our fusion models are put to the test on three different multi-view counting data – sets: PETS2009, DukeMTMC, and a freshly obtained dataset that contains a crowded intersection. Compared to previous multi-view counting standards, the results achieved by these methods are state-of-the-art.

In 2021 [12], crowd counting will be necessary in a variety of scenarios where it has traditionally been conducted using approximate (manual) estimates and measurements. If we use deep learning, we can fix this problem. Recent crowd counting methods typically utilize deep convolutional neural networks (CNNs) with tens of millions of parameters to create pixel-wise density maps. These models necessitate high-performance GPUs for training, inference and usage. Since smart devices like surveillance camera, mobiles, and Internet of things devices have limited processing capabilities, it is challenging to distribute these models to them. This work provides a novel approach to this problem with three essential components: feature fusion, Bayesian Loss, and datasets with bounding-box annotations to improve the efficiency of the crowd counting task.In order to improve the effectiveness of the crowd counting task, this study suggests a new approach based on three essential aspects: feature fusion, Bayesian Loss, and datasets making use of bounding-box annotations. According to the results of the experiments, the suggested method may not only allow real time in edge devices with limited processing capability, but also deliver accuracy comparable to the most recent deep learning algorithms.

In 2020 [13],many places still use old-fashioned ways to count crowds, such as keeping registers, using people counters, and using sensors at the entrance. The areas with fully random, highly variable, and dynamic human movement are not suitable for these techniques. In addition to being laborious, these procedures take a lot of time. The proposed method was created for times when rapid evacuation is necessary, such as during fires, natural disasters, and other similar scenarios, as well as making intelligent decisions based on the amount of people, such as food, water, congestion detection, etc. A system based on a deep convolutional neural network (DCNN) can be utilized for near-real-time crowd counting. The system utilizes the NVIDIA GPU processor and the parallel computing framework to provide rapid and agile processing of a camera's video feed. This study helps build a CCTV head detection model. The model is trained using overlapping heads, partial head visibility, etc. This technique accurately estimates headcount in dense populations in less time. This technique accurately estimates headcount in dense populations in less time.

In 2020 [14], major events have occurred in our world recently that have brought more attention to the significance of automatic crowd scene analysis. When a large number of people congregate in one place, as in the case of the COVID-19 breakout or a public event, it is necessary to have an automated system in place to monitor the area's inhabitants and ensure their safety. Heavy occlusion, complex behaviors, and changes in posture make analyzing crowd scenes extremely difficult. This research examines approaches for understanding congested scenes based on deep learning. The studied methods are divided into two categories: (1) crowd counting and (2) crowd action recognition. furthermore, databases of crowd scenes are investigated. In addition to the surveys mentioned previously, this research presents an evaluation score for crowd scene analysis techniques. In crowd scene videos, this measure estimates the discrepancy between the calculated and actual crowd counts.

3 Dataset description

In proposed work used dataset was download from internet. Dataset named (Human Detection Dataset) and download link is: https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset This dataset consists of two classes which are (no humans, and humans). Dataset consists of 921 images (indoor, and outdoor).

a. No human class contains 362, Figure 1 illustrate samples of no human class.

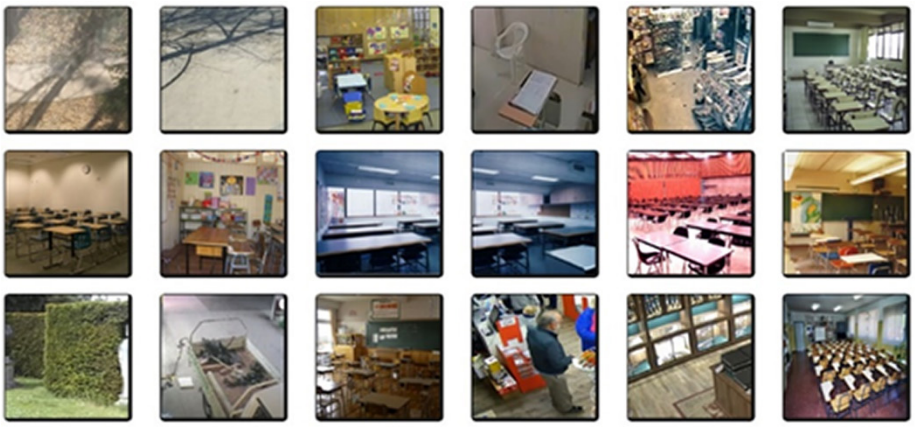

Fig. 1. Samples of No-Human class

b. human class contains 559 images. Figure 2 illustrate samples of human class.

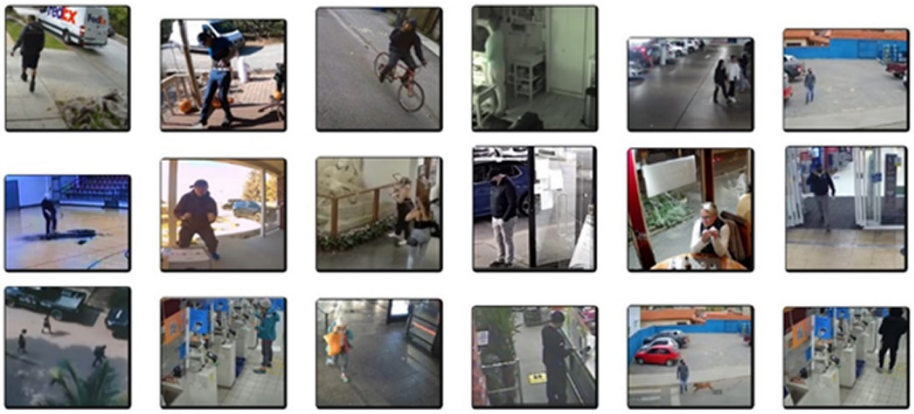

Fig. 2. Samples of human class

In proposed work split the dataset into two parts (70% of dataset for training, and 30% for validation).

4 YOLOv5 Architectures

YOLOv5 algorithm is trained on the (human detection dataset). At first dataset was preparing by annotation human objects from images of (human class) then train algorithm. In this paper used YOLOv5 architecture consists of 24 layers in additional to detection layer, Figure 3 illustrate YOLOv5 architecture.

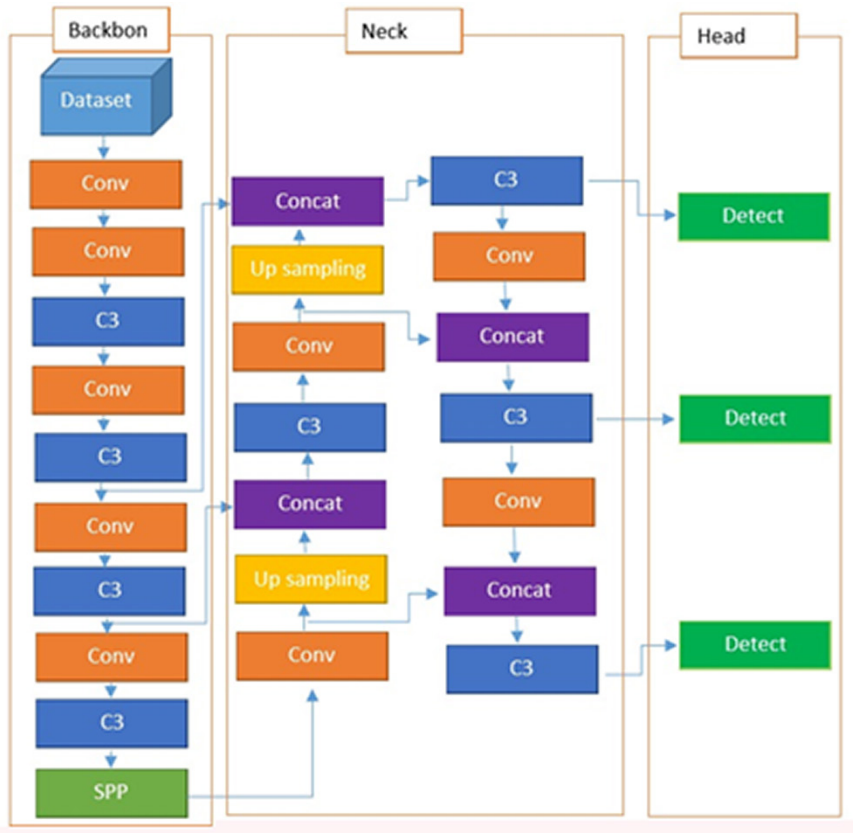

Fig. 3. YOLOv5 architecture

Four models of a YOLOv5 algorithm were built which are:

4.1 YOLOv5s

This model consists of 25 layers and in each layer the image is scaled in addition to extracting the features using certain mask sizes. Table 1 illustrate algorithm layers and specification.

Table 1. YOLOv5s layers and specifications

Part	Layer	Specifications
Backbone	Conv	[3,32,6,2,2]
	Conv	[32,64,3,2]
	C3	[64,64,1]
	Conv	[64,128,3,2]
	C3	[128,128,2]
	Conv	[128,256,3,2]
	C3	[256,256,3]
	Conv	[256,512,3,2]
	C3	[512,512,1]
	SPP	[512,512,5]
Neck	Conv	[512,256,1,1]
	Upsampling	[2,'nearest']
	Concat	[1]
	C3	[512,256,1]
	Conv	[256,128,1,1]
	Upsampling	[2,'nearest']
	Concat	[1]
	C3	[256,128,1]
	Conv	[128,128,3,2]
	Concat	[1]
	C3	[256,256,1]
	Conv	[256,256,3,2]
	Concat	[1]
	C3	[512,512,1]
Head	Detect	[1]

4.2 YOLOv5m

This model consists of 25 layers also but have different specifications of layers. Table 2 illustrate algorithm layers and specification.

Table 2. YOLOv5 m layers and specifications

Part	Layer	Specifications
Backbone	Conv	[3,48,6,2,2]
	Conv	[48,96,3,2]
	C3	[96,96,2]
	Conv	[96,192,3,2]
	C3	[192,192,4]
	Conv	[192,384,3,2]
	C3	[384,384,6]
	Conv	[384,768,3,2]
	C3	[768,768,2]
	SPP	[768,768,5]
Neck	Conv	[768,384,1,1]
	Upsampling	[2, 'nearest']
	Concat	[1]
	C3	[768,384,2]
	Conv	[384,192,1,1]
	Upsampling	[2, 'nearest']
	Concat	[1]
	C3	[384,192,2]
	Conv	[192,192,3,2]
	Concat	[1]
	C3	[384,384,2]
	Conv	[384,384,3,2]
	Concat	[1]
	C3	[768,768,2]
Head	Detect	[1]

4.3 YOLOv5l

This model consists of 25 layers. Table 3 illustrate algorithm layers and specification.

Table 3. YOLOv5l layers and specifications

Part	Layer	Specifications
Backbone	Conv	[3,32,6,2,2]
	Conv	[32,64,3,2]
	C3	[64,64,1]
	Conv	[64,128,3,2]
	C3	[128,128,2]
	Conv	[128,256,3,2]
	C3	[256,256,3]
	Conv	[256,512,3,2]
	C3	[512,512,1]
	SPP	[512,512,5]
Neck	Conv	[512,256,1,1]
	Upsampling	[2,'nearest']
	Concat	[1]
	C3	[512,256,1]
	Conv	[256,128,1,1]
	Upsampling	[2,'nearest']
	Concat	[1]
	C3	[256,128,1]
	Conv	[128,128,3,2]
	Concat	[1]
	C3	[256,256,1]
	Conv	[256,256,3,2]
	Concat	[1]
	C3	[512,512,1]
Head	Detect	[1]

4.4 YOLOv5x

This model consists of 25 layers and in each layer the image is scaled in addition to extracting the features using certain mask sizes. Table 4 illustrate algorithm layers and specification.

Table 4. YOLOv5x layers and specifications

Part	Layer	Specifications
Backbone	Conv	[3,32,6,2,2]
	Conv	[32,64,3,2]
	C3	[64,64,1]
	Conv	[64,128,3,2]
	C3	[128,128,2]
	Conv	[128,256,3,2]
	C3	[256,256,3]
	Conv	[256,512,3,2]
	C3	[512,512,1]
	SPP	[512,512,5]
Neck	Conv	[512,256,1,1]
	Upsampling	[2,'nearest']
	Concat	[1]
	C3	[512,256,1]
	Conv	[256,128,1,1]
	Upsampling	[2,'nearest']
	Concat	[1]
	C3	[256,128,1]
	Conv	[128,128,3,2]
	Concat	[1]
	C3	[256,256,1]
	Conv	[256,256,3,2]
	Concat	[1]
	C3	[512,512,1]
Head	Detect	[1]

5 Experimental results

For the purpose of testing the proposed models of the YOLOv5 algorithm, a (human detection) dataset was used and the models was trained to detect people and comparison between results of each model and as a following:

A. YOLOv5l: the accuracy results of this model arrived to 96.41% for training data, and 97.54% for validation data. Figure 4 illustrate results of YOLOv5l model.

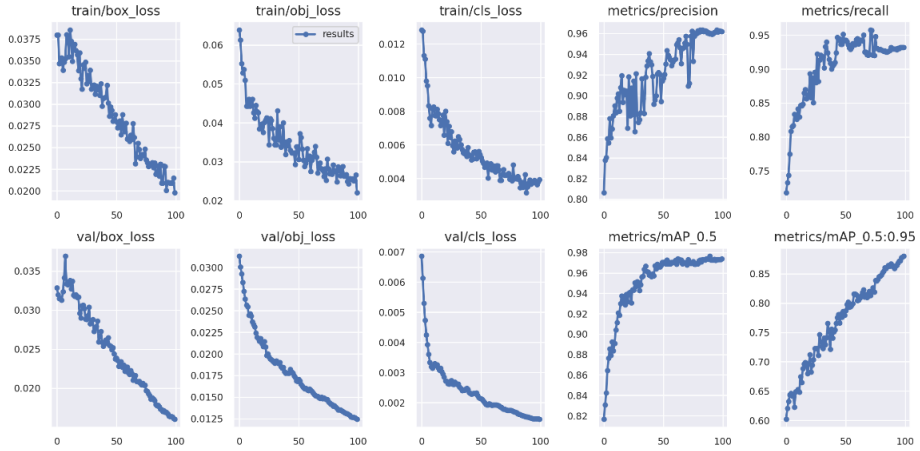

Fig. 4. Results of YOLOv5l model

B. YOLOv5m: the accuracy results of this model arrived to 90.96% for training data, and 97.51% for validation data. Figure 5 illustrate results of YOLOv5m model.

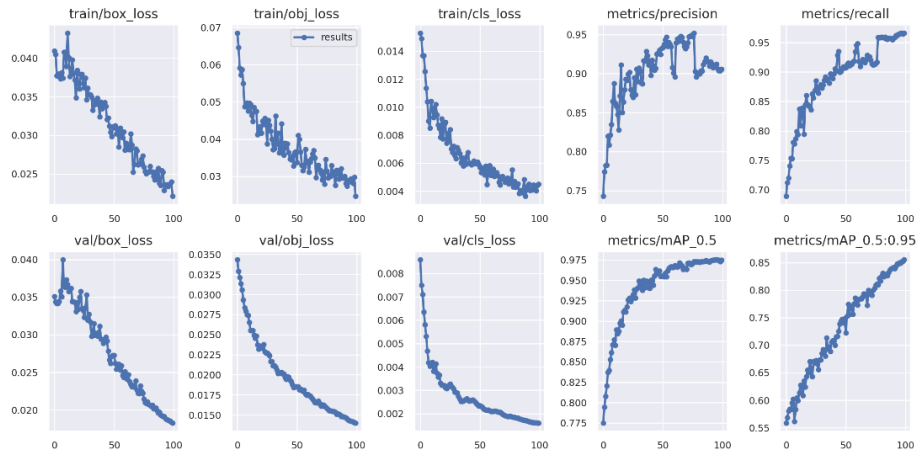

Fig. 5. Results of YOLOv5m model

C. YOLOv5s: the accuracy results of this model arrived to 93.22% for training data, and 95.14% for validation data. Figure 6 illustrate results of YOLOv5s model.

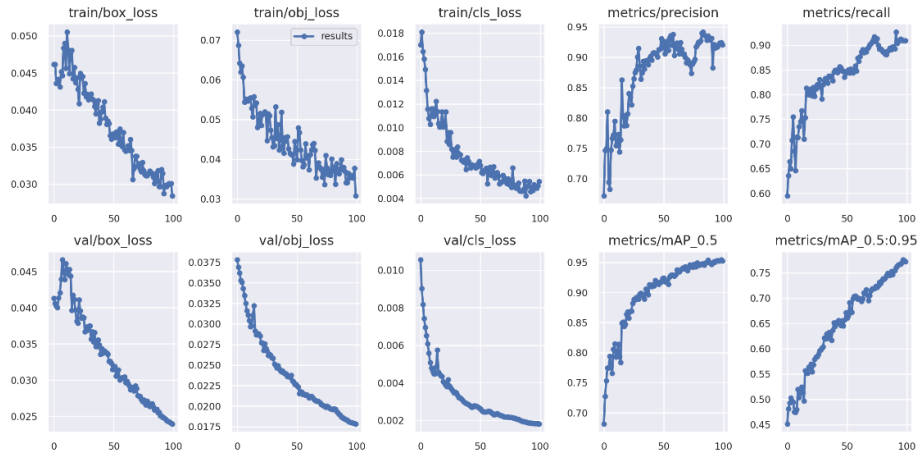

Fig. 6. Results of YOLOv5s model

D. YOLOv5x: the accuracy results of this model arrived to 96.53% for training data, and 97.56% for validation data. Figure 7 illustrate results of YOLOv5l model.

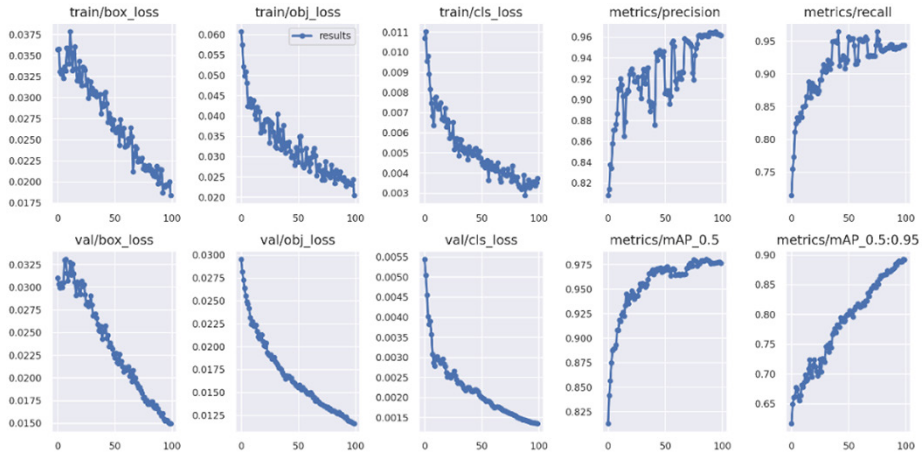

Fig. 7. Results of YOLOv5x model

Through previous experiments, conclude that the YOLOv5x model has reached a higher accuracy than the rest in training and testing, and the results can be summarized in Table 5.

Table 5. Summary of models results

	YOLOv5l	**YOLOv5m**	**YOLOv5s**	**YOLOv5x**
Train box loss	0.0189	0.0213	0.0269	0.0183
Train object loss	0.023	0.023	0.031	0.020
Train CLS loss	0.0041	0.0048	0.0588	0.0038
Precision	0.9641	0.9096	0.9322	0.9653
Recall	0.9354	0.9454	0.9114	0.9493
Validation box loss	0.0127	0.0221	0.0235	0.0150
Validation object loss	0.0013	0.0143	0.0175	0.0115
Validation CLS loss	0.0017	0.0019	0.0018	0.0013
mAP_0.5	0.9754	0.9751	0.9514	0.9756
mAP_0.5: 0.95	0.8835	0.8639	0.7854	0.8971

6 System testing

For the purpose of evaluating the system, a database called (**Crowd-UIT**) was used. This database of data contains ten videos that were taken in different places, and it was concluded that the system achieved accuracy in identifying and counting people up to more than 91 percent. Figure 8 illustrate samples of proposed system crowd detection, Table 6 illustrate system evaluation for all dataset videos.

Fig. 8. Crowd detection results

Table 6. Proposed system evalution

Video	Size	Number of Person in Video	Number of Detection	Accurecy
1	1280 × 720	79	74	0.9367
2	1280 × 720	65	61	0.9385
3	1280 × 720	224	207	0.9241
4	1920 × 1080	565	543	0.9611
5	1920 × 1080	570	523	0.9175
6	1920 × 1080	207	186	0.8986
7	1920 × 1080	114	102	0.8947
8	1920 × 1080	170	158	0.9294
9	1280 × 720	72	67	0.9305
10	1280 × 720	72	65	0.9028

7 Conclusions

There are many conclusions that the research reached through many practical experiments that were conducted on the subject of detecting people, especially in crowded places. During this paper, the YOLOv5 family was tested, during which four models of the YOLOv5 algorithm were trained for the purpose of measuring the accuracy of each model in detecting people. From these tests can conclude that the highest accuracy reached by the model YOLOv5x, where this model reach accuracy 96.53 in training and 97.56 in testing as Table 5. The reason that the accuracy of the model is more than the rest of the models is that the sizes of the filters were very suitable for identifying people when trying them on photos or videos. When using a dataset (**Crowd-UIT**) which contains ten videos, the accuracy of identifying people has reached 92.34% as in Table 6.

8 Acknowledgments

The authors would like to thank Mustansiriyah University in Bagdad, Iraq, for their cooperation with this study (http://uomustansiriyah.edu.iq).

9 References

[1] W. M. Salih, I. Nadher, and A. Tariq, "Deep learning for face expressions detection: Enhanced recurrent neural network with long short term memory," in *Applied Computing to Support Industry: Innovation and Technology: First International Conference, ACRIT 2019, Ramadi, Iraq, September 15–16, 2019, Revised Selected Papers*, 2020: Springer, pp. 237–247. https://doi.org/10.1007/978-3-030-38752-5_19

[2] 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)*, 2017: IEEE, pp. 1–7. https://doi.org/10.1109/AVSS.2017.8078482

[3] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 640–644. https://doi.org/10.1145/2964284.2967300

[4] 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. https://doi.org/10.1109/CVPR.2019.00524

[5] W. M. S. Abedi, D. Ibraheem-Nadher, and A. T. Sadiq, "Modified Deep Learning Method for Body Postures Recognition," *International Journal of Advanced Science and Technology*, vol. 29, no. 2, pp. 3830–3841, 2020.

[6] A. S. Hussein, R. S. Khairy, S. M. M. Najeeb, and H. T. ALRikabi, "Credit card fraud detection using fuzzy rough nearest neighbor and sequential minimal optimization with logistic regression," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 5, 2021. https://doi.org/10.3991/ijim.v15i05.17173

[7] A. Al-zubidi, R. K. Hasoun, and S. H. Hashim, "Mobile application to detect covid-19 pandemic by using classification techniques: Proposed system," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 16, pp. 34–51, 2021. https://doi.org/10.3991/ijim.v15i16.24195

[8] S. H. Abbood, H. N. Abdull Hamed, M. S. Mohd Rahim, and A. H. M. Alaidi, "DR-LL Gan: Diabetic retinopathy lesions synthesis using generative adversarial network," *International Journal of Online & Biomedical Engineering*, vol. 18, no. 3, 2022. https://doi.org/10.3991/ijoe.v18i03.28005

[9] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017: IEEE, pp. 465–469. https://doi.org/10.1109/ICIP.2017.8296324

[10] S. Huang *et al.*, "Body structure aware deep crowd counting," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1049–1059, 2017. https://doi.org/10.1109/TIP.2017.2740160

[11] Q. Zhang and A. B. Chan, "Wide-area crowd counting: Multi-view fusion networks for counting in large scenes," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1938–1960, 2022. https://doi.org/10.1007/s11263-022-01626-4

[12] Z. Huang, R. Sinnott, and Q. Ke, "Crowd counting using deep learning in edge devices," in *2021 IEEE/ACM 8th International Conference on Big Data Computing, Applications and Technologies (BDCAT'21)*, 2021, pp. 28–37. https://doi.org/10.1145/3492324.3494161

[13] U. Bhangale, S. Patil, V. Vishwanath, P. Thakker, A. Bansode, and D. Navandhar, "Near real-time crowd counting using deep learning approach," *Procedia Computer Science*, vol. 171, pp. 770–779, 2020. https://doi.org/10.1016/j.procs.2020.04.084

[14] S. Elbishlawi, M. H. Abdelpakey, A. Eltantawy, M. S. Shehata, and M. M. Mohamed, "Deep learning-based crowd scene analysis survey," *Journal of Imaging*, vol. 6, no. 9, p. 95, 2020. https://doi.org/10.3390/jimaging6090095

10 Authors

Mohammed Abdul Jaleel Maktoof, Computer Science Department, University of Technology, Baghdad, Iraq.

Israa Tahseen Ali Al_attar, Computer Science Department, University of Technology, Baghdad, Iraq.

Ibraheem Nadher Ibraheem, Faculty of Basic Education, Mustansiriyah University, Baghdad, Iraq.

Article submitted 2023-01-13. Resubmitted 2023-02-23. Final acceptance 2023-02-24. Final version published as submitted by the authors.