Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2023, 2023:103
https://doi.org/10.28919/cmbn/8097
ISSN: 2052-2541

EVALUATION OF CROWD COUNTING MODELS IN TERM OF PREDICTION PERFORMANCE AND COMPUTATIONAL REQUIREMENT

HENDRI SANTOSA, IGNATIUS HANSEN, GEDE PUTRA KUSUMA[*]

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Abstract: With the increasing of human population and the development of technology, crowd counting models are needed to estimate people in certain areas. This research paper compares the prediction performance and computational requirement of four state of the art crowd counting models: M-SFAnet, DM-Count, Context-Aware Crowd Counting (ECAN), and Supervised Spatial Divide-and-Conquer (SS-DCNet). The evaluations were performed to find the most high-performance model in term of prediction performance and computational requirement. The computational requirement is being compared and considered because of the development of Internet of Things devices, crowd counting models that have good prediction performance and low computational requirements can be implemented in low-compute devices. We evaluated the models on four different datasets. From the evaluation we found that SS-DCNet approach achieved the most favorable results.

Keywords: crowd counting; models evaluation; deep learning; prediction performance; computational requirement.

2020 AMS Subject Classification: 92D25, 92D30.

1. INTRODUCTION

Crowd counting is a branch of science from crowd analysis that can be used to carry out monitoring and surveillance in video form, providing estimates in designing an area, monitoring

[*]Corresponding author
E-mail address: inegara@binus.edu

Received July 06, 2023

traffic, and others [1]. By using crowd count technology, the total mass and mass density values can be estimated [2]. Research on crowd counting is highly important for security purposes. A recent tragedy caused by the crowd is the Seoul tragedy, with at least 153 people killed. Nowadays, many crowd counting models are being developed, but many of them are focused on prediction performance. Several crowd counting models that are close, if not the best, in the state of the art in terms of prediction performance are M-SFANet, DM-Count, Context-Aware Crowd Counting (ECAN), and Supervised Spatial Divide-and-Conquer Network (SS-DCNet). Frequently when evaluating a crowd counting model the focus is on finding the least prediction error. On the other hand, the computational requirement is ignored. The computational requirement is equally important to discuss to understand how heavy the model and the possibility of the application of the model. The high demand of light-weight model due to the development of the Internet of Things (IoT) also a reason why the efficiency of a model is important. Therefore, in this research from the four models mentioned above, the computational requirements and prediction performance of each model are being analyzed and evaluated. Therefore, the result and comparison of its computational requirement and prediction performance can then be shown.

2. Related Works

Early approach of crowd counting is by detecting individuals in the input picture and then counting the detected individuals. The detection is done by bounding boxes that slide to find the desired object which in this case is persons (i.e., object detection approach). This approach was later found inefficient and heavy in the computing sense, since it requires the model accurately detects the individual in a crowded situation which makes individuals often overlap and make it hard for the model correctly detects every individual on the scene. In attempts to reduce the computational weight of the detection, the “object” that will be detected by the model is reduced (in features) to just the head. This attempt is still not enough to reduce the computation. One of the attempt of crowd counting by detection proposed by Marsden *et al.* [3] that build a model based on Resnet18 network of He *et al.* [4], which is trained on ImageNet dataset. Feature map average pooling step was used in the model to reduce the parameters, hence allowing multi-task crowd analysis to be applied and reduces the memory needed in the training process. The model was also tasked to detect violent behavior which justify the chosen approach of detection that can both detect violent behavior and count the crowd. Other attempt is by Xing *et al.* [5], who proposed crowd counting based on detection flows. By tackling crowd counting with detection flows, they can reduce false alarms, can work better with data noises, and give better specific descriptions of crowds.

Then the trend shifted to regression-based methods. This approach to some degree successfully deals with dense crowd's situations and high background clutter. This approach is inspired by the human ability to estimate the density of a crowd at first glance without counting individually the crowd. The regression approach of crowd counting is achieved by determining crowd density from low-level imagery features. First global feature (such as texture and edges) and local feature (such as Scale-invariant Feature Transform (SIFT), Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gray level Co-occurrence Matrix (GLCM)) are extracted from the image [6]. After the feature extraction, regression models are trained to predict the number of people in the crowd. It found out that local feature is the best feature to use in Regression method of crowd counting compared to holistic features and histogram features. Also, the Gaussian process regression is the best regression for crowd counting compared to linear regression, k-nearest neighbors and neural networks [7].

At later stage, arose a new approach along with the breakthrough of Deep Learning models that excels at tackling Computer Vision problems [8]. The proposed method suggested to utilize a Deep Learning model like CNN to crowd counting. The approach is combined with the concept of extracting features from the regression approach. But instead of the regression model, the Deep Learning model is used to predict the crowd counts. This kind of approach by far surpasses past approaches.

In addition to the Deep Learning approach, the current trend is to extract a density map from the image for features that will be fed to the model. This reduces the computational weight and relieves the data from the less useful feature. The model now does not need to recognize complex features, it only needs to recognize simple features from the image which now is in the form of the density map.

3. THEORY AND METHODS

3.1. M-SFANET

The first model is M-SFANet. This model consists of four main components: VGG16-bn as the encoder, which constantly reduces the feature map size and captures high-level semantics information, the multi-scale-aware modules that are divided into Context-aware module (CAN), which is connected with the 10th layer of VGG-16bn, Atrous spatial pyramid pooling (ASPP), which is connected to the 13th layer of VGG-16bn, and lastly the dual path multi-scale fusing decoder consists of density and attention map path [9]. The process of M-SFANet starts when the input image is fed to the encoder to become a feature map, then the feature map is fed into the multi-scale-aware modules (CAN and ASPP), and finally the decoder will combine the multi-scale feature into a density and attention maps [10].

3.2. DM-Count

The second model is DM-Count. This model uses VGG-19 as its backbone. DM-Count performs Distribution Matching to do crowd counting. Optimal Transport (OT) is used to measure the similarity between the predicted density map and the ground truth. Total Variation (TV) is also used to stabilize the OT. In DM-Count, the Gaussian method was not used because the Gaussian method will impact the generalization performance of crowd counting [11].

3.3. Context Aware Crowd Counting (ECAN)

The third model is Context-Aware Crowd Counting (ECAN). The ECAN approach is a deep net architecture that adaptively encodes multi-level contextual information into features. The ECAN approach is meant to overcome the large-scale consistencies that appear in images. ECAN uses the first ten layers of pre-trained VGG-16 as its backbone then by performing Spatial Pyramid Pooling, the scale-aware features are computed. Spatial Pyramid Pooling is used to extract multi-scale context from VGG. Then the geometry of the images being exploited, this addressed to cover the multi geometry of the images across it vary. The strategy used to determine the ground-truth density maps is to denote each position of the human head in the scene and by convolving an image. To minimize the loss, Stochastic Gradient Descend (SGD) and Adam algorithm are used [12].

3.4. Supervised Spatial Divide-and-Conquer Network (SS-DCNet)

The fourth model is SS-DCNet. This model used pre-trained VGG-16 as the encoder [13] and U-net [14] as the decoder to obtain the feature map. Then the first stage of Spatial Devide-And-Conquer (S-DC) is executed to fuse the feature map. SS-DCNet can also execute multi-stage S-DC by doing further decoding. There are multiple loss functions that SS-DCNet used, which are Counter Loss, Merging Loss, Division Loss, Upsampling Loss, and Division Consistency Loss, and then the ground truth can be obtained [15].

4. Proposed Methods

4.1. Dataset

Four datasets are used in this research, namely ShanghaiTech Part A [16], ShanghaiTech Part B [16], UCF_CC_50 [17], and UCF_QNRF [18].

TABLE 1. Used Datasets Quantity.

Dataset	Train Dataset	Test Dataset
ShanghaiTech Part A	300	182
ShanghaiTech Part B	400	316
UCF_CC_50	-	50
UCF_QNRF	1201	334

4.2. Model

Four models are used to implement the four datasets, namely MSFA-Net, DM-Count, Context-Aware Crowd Counting (ECAN), and Supervised Spatial Divide-And-Conquer Network (SS-DCNet). Each model is going to be compiled into four datasets as mentioned in Table 1.

4.3. Evaluation of Prediction Performance

The experiment evaluates the prediction performance based on the value of Mean Square Error (MSE).

$$MSE = \frac{\sum(yi - \widehat{yi})^2}{n} \qquad (1)$$

Where *yi* is the *i* observed value, *(yi)* ˆis the *i* predicted value, and *n* is the number of observations and MAE (Mean Absolute Error)

$$MAE = \frac{\sum |yi - xi|}{n} \qquad (2)$$

Where *yi* is the predicted value, *xi* is the observed value, and *n* is the number of observations. The experiment is tested using the test data from each dataset as shown in Table 1 and is run in each pre-trained model to obtain the MSE and MAE result.

4.4. Evaluation of Computational Requirement.

The experiment evaluates the computational requirement based on the average CPU usage, average RAM usage, average runtime of each image in one specific dataset, and the average runtime of each dataset. CPU usage and RAM usage is being observed in the task manager while the average runtime of each image and average runtime of each dataset is collected based on the time of compilation in each dataset. To obtain the average value, pre-trained models are run on test data five times in each dataset.

TABLE 2. Hardware Specification.

CPU	RAM size
AMD Ryzen 5 4600H	16 GB of DDR4

4.5. Overall Performance Evaluation.

The last step is to determine from the experiment results which model is the highest performing (both highly accurate and fast) to do crowd counting using the hardware stated in Table 2. The way to find the best high-performing model is by rank each model by performance (evaluation score: MAE and MSE) and Computational Requirement (Mainly the runtime), then add both rank and find which is the lowest. If the tie result found, the most efficient model is the one that have more balance result between the accuracy rank and computational requirement rank.

5. RESULT AND DISCUSSION

5.1. Evaluation Result of Prediction Performance.

Table 3. Evaluation of Prediction Performance on Test Dataset ShanghaiTech Part A.

ShanghaiTech Part A	MAE	MSE
MSFA-Net	**57.55**	**94.48**
DM-Count	59.7	148.3
SSDC-Net	58.3	95
ECAN	62.3	100

Table 4. Evaluation of Prediction Performance on Test Dataset ShanghaiTech Part B.

ShanghaiTech Part B	MAE	MSE
MSFA-Net	**6.32**	**10.06**
DM-Count	7.4	11.8
SSDC-Net	6.7	10.7
ECAN	7.8	12.2

Table 5. Evaluation of Prediction Performance on Test Dataset UCF_CC_50.

UCF_CC_50	MAE	MSE
MSFA-Net	**162.33**	**276.76**
DM-Count	211	291.5
SSDC-Net	204.2	301.3
ECAN	212.2	243.7

Table 6. Evaluation of Prediction Performance on Test Dataset UCF_QNRF.

UCF_QNRF	MAE	MSE
MSFA-Net	85.60	151.23
DM-Count	**85.60**	**148.3**
SSDC-Net	104.4	176.1
ECAN	107	183

As Shown in Table 3, 4, and 5, MSFA-Net model has the best prediction performance score on ShanghaiTech Part A test dataset with 57.55 for MAE score and 94.48 for MSE score, on ShanghaiTech Part B test dataset with 6.32 for MAE score and 10.06 for MSE score, and UCF_CC_50 test dataset with 162.33 for MAE score and 276.76 for MSE score. However, on UCF_QNRF test dataset, DM-Count model produces the best evaluation of prediction performance with 85.60 for its MAE score and 148.3 for its MSE score as shown in Table 6. Overall, MSFA-Net model has the has the best prediction performance in three of four tested datasets.

5.2. Evaluation Result of Computational Requirement

Table 7. Evaluation of Computational Performance on Test Dataset ShanghaiTech Part A.

ShanghaiTech Part A	Average Runtime per Image (s)	Average Runtime (s)	Average CPU Usage (%)	Average RAM Usage (% of 16 GB)
MSFA-Net	2.15	395.28	72.56	4.57
DM-Count	**1.44**	**266.19**	**69.34**	**4.47**
SSDC-Net	1.61	296.7	72.84	4.6
ECAN	1.9	347.1	71.54	4.58

Table 8. Evaluation of Computational Performance on Test Dataset ShanghaiTech Part B.

ShanghaiTech Part B	Average Runtime per Image (s)	Average Runtime (s)	Average CPU Usage (%)	Average RAM Usage (% of 16 GB)
MSFA-Net	3.2	1017.22	72.00	3.74
DM-Count	**2.15**	**687.5**	**69.82**	**4.19**
SSDC-Net	2.25	718.98	73.16	4.09
ECAN	2.8	887.1	72.04	3.77

Table 9. Evaluation of Computational Performance on Test Dataset UCF_CC_50.

UCF_CC_50	Average Runtime per Image (s)	Average Runtime (s)	Average CPU Usage (%)	Average RAM Usage (% of 16 GB)
MSFA-Net	2.45	124.53	71.12	4.35
DM-Count	**1.71**	**86.91**	**71.14**	**4.37**
SSDC-Net	-	-	-	-
ECAN	2.15	107.61	70.04	4.4

Table 10. Evaluation of Computational Performance on Test Dataset UCF_QNRF.

UCF_QNRF	Average Runtime per Image (s)	Average Runtime (s)	Average CPU Usage (%)	Average RAM Usage (% of 16 GB)
MSFA-Net	30.53	10199.44	74.00	47.94
DM-Count	11.03	3717.15	70.50	23.35
SSDC-Net	**5.56**	**1874.72**	**73.46**	**9.08**
ECAN	22.65	7567.06	72.13	49.38

As shown in Table 7, 8, and 9, DM-Count model has the best computational performance. On ShanghaiTech Part A test dataset DM-Count model produce average runtime/images score of 1.44s, average runtime score of 266.19s, average CPU usage of 69.34%, and average RAM usage score of 4.47%. On ShanghaiTech Part B test dataset DM-Count produce average runtime/images score of 2.15s, average runtime score of 687.5, average CPU usage score of 69.82%, and average RAM usage score of 4.19%. On UCF_CC_50 test dataset DM-Count produces average runtime/images score of 1.71s, average runtime score of 86.91s, average CPU usage score of 71.14%, and average

RAM usage score of 4.37%. However, on UCF_QNRF dataset SSDC-Net model has the lowest overall computational performance with average runtime/images score of 5.56s, average runtime score of 1874.72s, average CPU usage of 73.46%, and average RAM usage of 9.08% as shown in Table 10. Overall, DM-Count model has the best computational performance on three of four tested datasets.

5.3. Overall Performance Evaluation Result.

Table 11. Evaluation of Prediction Performance on Test Dataset UCF_QNRF.

Model	Accuracy Rank (MAE and MSE)	Computational Requirement Rank	Overall Performance Score (the lower the better)
MSFA-Net	1	4	5
DM-Count	3	1	4
SSDC-Net	**2**	**2**	**4**
ECAN	4	3	7

From the Performance Evaluation Result as shown in Table 11, it is found that SSDC-Net is the most high-performing followed by DM-Count on second, MSFA-Net on third, and ECAN on the least high-performing of all. SSDC-Net and DM-Count have the same Performance Overall Performance Score, but since SSDC-Net got a more balanced rank between Performance and Computational Performance, it is decided that SSDC-Net is the better high-performing one. DM-Count is the most efficient approach but came third on accuracy. MSFA-Net approach got first on Performance rank but is by far the slowest approach. ECAN got fourth in accuracy rank and third in computational requirement rank which makes it the least high-performing approach.

6. CONCLUSIONS AND FUTURE WORK

In this research, the prediction performance and computational requirement are compared and evaluated in four different crowd counting models. Based on the result, MSFA-Net has the lowest MAE and MSE value in three of the four datasets that are being tested which make it the most effective approach, yet the model is heavier than the other, ranked last in runtime. On the other hand, DM-Count has the lowest average CPU usage, average RAM usage, average runtime of each image in one specific dataset, and the average runtime of each dataset which makes it the most efficient approach. The result also shows that SSDC-Net is the best high-performing model (comes second on both accuracy rank and computational performance Rank) since it is both highly accurate and fast.

In future works, this research can be a reference for the researcher to develop a crowd counting model that's not only focused on prediction performance but also focused on the computational requirement of the crowd counting model.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] Q. Wang, J. Gao, W. Lin, 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 (CVPR), 2019, pp. 8198-8207.

[2] J. Sang, W. Wu, H. Luo, et al. Improved crowd counting method based on scale-adaptive convolutional neural network, IEEE Access. 7 (2019), 24411-24419. https://doi.org/10.1109/access.2019.2899939.

[3] M. Marsden, K. McGuinness, S. Little, et al. 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, Lecce, Italy, 2017: pp. 1-7. https://doi.org/10.1109/AVSS.2017.8078482.

[4] W. Li, V. Mahadevan, N. Vasconcelos, Anomaly detection and localization in crowded scenes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2013), 18-32. https://doi.org/10.1109/tpami.2013.111.

[5] J. Xing, H. Ai, L. Liu, et al. Robust crowd counting using detection flow, in: 2011 18th IEEE International Conference on Image Processing, IEEE, Brussels, Belgium, 2011: pp. 2061-2064.
https://doi.org/10.1109/ICIP.2011.6115886.

[6] K. Chen, C.C. Loy, S. Gong, et al. Feature mining for localised crowd counting. In: Proceedings of the British Machine Vision Conference (BMVC), p. 3 (2012).

[7] D. Ryan, S. Denman, S. Sridharan, et al. An evaluation of crowd counting methods, features and regression models, Computer Vision Image Understand. 130 (2015), 1-17. https://doi.org/10.1016/j.cviu.2014.07.008.

[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM. 60 (2017), 84-90. https://doi.org/10.1145/3065386.

[9] L. Zhu, Z. Zhao, C. Lu, et al. Dual path multi-scale fusion networks with attention for crowd counting, preprint, (2019). http://arxiv.org/abs/1902.01115.

[10] P. Thanasutives, K. Fukui, M. Numao, et al. Encoder-decoder based convolutional neural networks with multi-scale-aware modules for crowd counting, in: 2020 25th International Conference on Pattern Recognition (ICPR),

IEEE, Milan, Italy, 2021: pp. 2382-2389. https://doi.org/10.1109/ICPR48806.2021.9413286.

[11] B. Wang, H. Liu, D. Samaras, et al. Distribution matching for crowd counting, in: 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, 2020.

[12] W. Liu, M. Salzmann, P. Fua, Context-aware crowd counting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5099-5108.

[13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, (2015). http://arxiv.org/abs/1409.1556.

[14] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, Springer International Publishing, Cham, 2015: pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28.

[15] H. Xiong, H. Lu, C. Liu, et al. From open set to closed set: supervised spatial divide-and-conquer for object counting, (2020). http://arxiv.org/abs/2001.01886.

[16] Y. Zhang, D. Zhou, S. Chen, et al. Single-image crowd counting via multi-column convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 589-597.

[17] A. Bansal, K.S. Venkatesh, People counting in high density crowds from still images, preprint, (2015). http://arxiv.org/abs/1507.08445.

[18] H. Idrees, M. Tayyab, K. Athrey, et al. Composition loss for counting, density map estimation and localization in dense crowds, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 532-546.