Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 770–779

Procedia Computer Science

www.elsevier.com/locate/procedia

-Third International Conference on Computing and Network Communications (CoCoNet’19)

Near Real-time Crowd Counting using Deep Learning Approach

Ujwala Bhangale, Suchitra Patil, Vaibhav Vishwanath, Parth Thakker, Amey Bansode, Devesh Navandhar

Department of Information Technology
K.J. Somaiya College of Engineering Vidyavihar, Mumbai, India

Abstract

In the current digital era, at many places crowd counting mechanisms still rely on old-fashioned methods such as maintaining registers, making use of people counters and sensors based counting at entrance. These methods fail in the places where the movement of people is completely random, highly variable and dynamic. These methods are time consuming and tedious. The proposed system is developed for situations where emergency evacuations are required such as fire outbreaks, calamitous events, etc. and making informed decisions on the basis of the number of people such as food, water, detecting congestion, etc. A deep convolution neural network (DCNN) based system can be used for near real-time crowd counting. The system uses NVIDIA GPU processor to exploit the parallel computing framework to achieve swift and agile processing of the video feed taken through a camera. This work contributes towards constructing a model to detect heads captured by CCTV camera. The model is trained extensively by providing several scenarios such as overlapping heads, partial visibility of heads etc. This system provides significant accuracy in estimating the head count in dense population in reasonably less amount of time.

Keywords: Multicore GPU processor, Deep Convolution Neural Network, Head-count, crowd counting

1. Introduction

Crowd counting is essential to serve many real-world applications, such as resource management (such as water, food supply), traffic control, security, disaster management etc. The traditional methods for crowd-counting such as

10.1016/j.procs.2020.04.084

manual counting, using registers to maintain records of each person, and counting through use of sensors are time consuming and tedious, and may produce fallible results due to dynamic movements. This has led to the evolution of crowd-counting methods which rely on CCTV video feeds. The major benefit of using counting methods on video feed is that the dynamism of people's movement cannot be incorporated in any of the previous ways of crowd counting. This requires a modern outlook into the problem. An accurate crowd counting system provides solutions for emergency situations such as fire outbreaks, earthquakes and many other disaster situations. In these conditions, an estimate of the crowd would allow the concerned authorities to make the correct decisions regarding supplies of resources. Recent solutions to the problem of an accurate estimate of crowd count have brought up techniques such as crowd counting using detection, density and regression [1][2][3].

In this paper, end-to-end solution is devised by providing a real time crowd counting mechanism for highly congested areas. It uses Very Deep Convolutional Neural Networks (DCNN) in its front-end to exploit its deep transfer learning and flexible configuration thus making it compatible with back end of Dilated Convolutional Neural Networks. The proposed systems ability to count the large crowds gathering of hundreds of people, deal with the complexity of partly visible head of a person, person wearing hats, etc. and also providing real time headcount of a video feed makes it stand out against its other competitions. This end-to-end system for crowd counting with video feed and processing input frames to produce the headcount is what separates this paper with others.

1.1. Previous work

Crowd counting and Analysis have a plethora of real-world applications such as planning emergency evacuations in case of fire outbreaks, calamitous events, etc. and making informed decisions on the basis of the number of people such as water, food planning, detecting congestion etc. and hence, there are many methods proposed to achieve crowd count [31].

I. Crowd counting based on object detection mechanism

Earlier approaches for crowd counting have used Detection Based methods. The detection methods are candid and it make use of off-the-shelf detectors [7][8][9] to detect the target objects and count these objects in images or videos.

1) Monolithic detection [10][11][12][13] which is called as the typical pedestrian approach where training is performed by extracting the human anatomy features. Some of the common features in this method are obtained using gradient-based features [14], edgelet [15], and shapelets [16]. The accuracy of human detection significantly depends on the classifier used for classification. This approach gives satisfactory detection in sparse scenes, but it does not work well in crowded scenes in presence of occlusion and clutter scene.
2) Part-Based Detection: By adopting a part-based detection method [17][18][19], there are several solutions to handle the partial occlusion problem till some extent; such as, head features are not sufficient to provide reliable results due to its shape variations, one can construct ensemble boosted classifier for specific body part considering its own set of features [20].
3) Multi-sensor detection: To resolve inter object occlusion which occurs due to the partial or full overlapping of more than one objects, the multi-view information from multiple cameras can be incorporated in the process. For instance, in [21], authors have estimated the crowd count by applying the multi-view geometric constraints to its full extent. Through the use of this methodology, the speed of detection is increased significantly apart from detection accuracy improvement. These supervised methods of crowd counting detect faces very well, but it fails in highly congested environments and even in surveillance applications where the resolution of the images affects its accuracy.

B. Regression based Crowd Counting

Some of the images are captured with low resolution, it is the major performance issue of detection-based crowd counting and the occluded multiple objects. Regression based counting performs better in this environment, where local features get extracted from the segmented images and then the regression model gets applied to estimates the crowd count in each segment [22][23][24]. Prior to this, regression-based methods were developed [25][26][27][28][29][31] using the global image features, but these approaches cannot capture the region wise distribution of the information. One of the crucial parts of this type of methods is extracting suitable features. This approach may overestimate the prediction when the crowd is less.

C. Crowd Counting by Density

The density-based methods generate density values which are estimated using low-level features such as pixels or regions, it overcomes the drawback of regression-based methods and also maintains the location information [31]. The predicted density maps may have different characteristics as the density map estimation methods may vary depending on the selection of the loss function and type of prediction. The prediction and loss function can either be region-wise or pixel-wise. Since image-wise prediction reuse computations, they are relatively faster. The insufficiency of these types of methods is that the actual count can often be inaccurate as mapping between density and image may lead to deviation.

2. Proposed DCNN based crowd counting approach

The main aim of the proposed work is to provide an end to end application for crowd counting through surveillance video feeds which is shown in figure1. This is achieved by running the crowd-counting algorithm on frames every second which allows to achieve near real-time processing in this system. In this section, a brief detail about the architecture of the proposed system is given along with the deep learning details for the crowd-counting algorithm.

A. Deep Convolution Neural Network (DCNN) Architecture

The DCNN architecture used for crowd counting is based on CSRNet [30]. As shown in figure 2, 10 convolution layers and 3 max pooling layers of VGG-16 [2] are used in the front end. In Back end, 6 dilated convolution layers with dilation rate of 2 is employed for optimal crowd-count results. The kernel size is maintained as 3X3 throughout.

Front-End Network: Many crowd counting research work employ the use of Very Deep Convolutional Neural Network (VGG 16) [2]. CrowdNet uses 13 of its convolution layers and adds a 1 X 1 convolution layer as output to achieve a density map [3], whereas the MultiColumn Convolutional Neural Network [1] uses VGG 16 as the density-level classifier to label the images before feeding it through on of its columns. However, all of these papers use VGG 16 as a utility and hence the final accuracy of the network is unaffected by it. The architecture mentioned in [30] uses the convolution layers of VGG 16 without its classification part (i.e. fully-connected layers). The output size of the image is 1/8th of the input size. This output is then fed into the dilated convolution back-end network which helps to maintain the resolution and generate high-quality density maps. This output is fed into the back-end dilated convolution network.

Back-End Network: The Dilated Convolutional Network is a crucial part of the architecture. A 2 dimensional dilated convolution can be defined as:

$$y(l,w) = \sum_{i=0}^{L} \sum_{j=1}^{W} x(l + r * i, w + r * j) \ldots\ldots (1)$$

here *y(l,w)* is the output of dilated convolution, *x(l,w)* is the input to the dilated convolution, a filter w(i,j) with length *L* and width *W* respectively and *r* is the dilation rate.

In literature, dilated Convolution is used for segmentation and observed as quite effective and accurate. Pooling layers i.e. max pooling and average pooling reduce the spatial distribution which makes the feature maps more distorted. We can suppress the effect of pooling layers by the addition of a deconvolution layer, but it results in an added complexity and latency problem, which is not suitable for real-time processing.

3) Network Configuration: The Neural Network uses VGG16 in its front-end and Dilated Convolution with a dilation rate of 2 instead of Pooling layers as the back end. The trade-off between accuracy and resource overhead (such as time for training, consumption of memory, and the number of parameters) has to be tailored to each

application separately. The fully-connected layers of VGG-16 are removed along with 2 pooling layers to play down the negative effects of pooling operation on output accuracy.

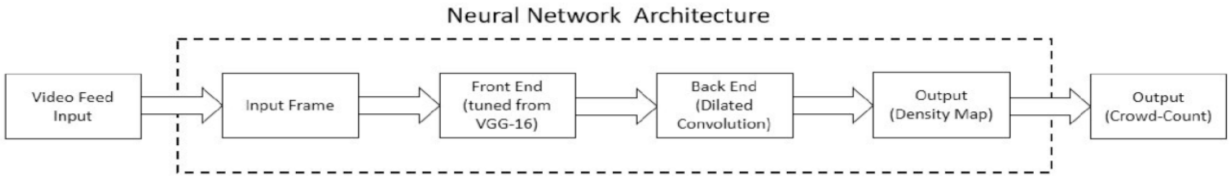

Neural Network Architecture
Video Feed Input
Input Frame
Front End (tuned from VGG-16)
Back End (Dilated Convolution)
Output (Density Map)
Output (Crowd-Count)

Figure 1. End-to-End Architecture for crowd counting

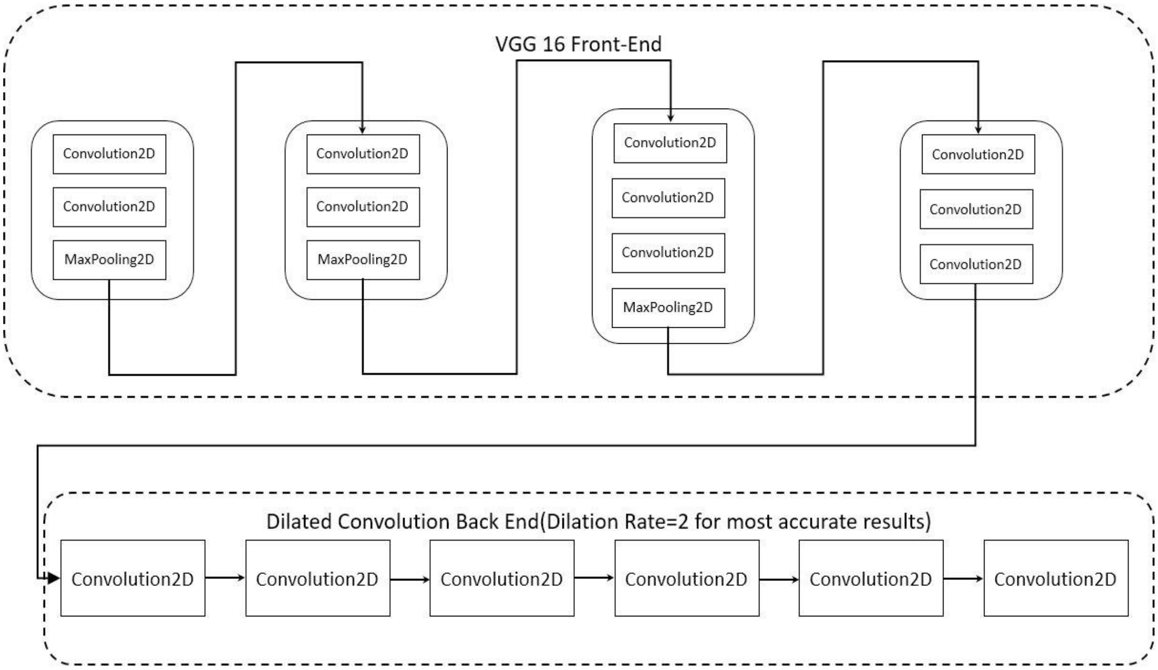


Figure 2. Neural Network Front End Back End based on CSRNet

IV. Training the network

The CSRNet [29] is trained on Keras-TensorFlow framework which gains a huge speedup in training and prediction using NVIDIA GPU parallel computing capability.

A. Training Aspects

CSRNet [29] is trained by considering the front-end and back end together as an end-to-end structure rather than individually training the front and back-end. A well-trained VGG-16 gives the first 10 fine-tuned layer of the network. During the training, stochastic gradient is applied and the learning rate is kept to 1e-6. Euclidean distance computed between the ground truth count and estimated crowd-count of the proposed methodology. The Loss is computed as,

$$L(\theta) = \frac{1}{2M}\sum_{i=1}^{M}||Y(X_i;\theta - Y_i^{GT}||_2^2 \dots\dots\dots.\dots(2)$$

where M is training set's size, $Y(X_i;\theta)$ is neural networks output Y_i^{GT} is the images actual ground truth.

Ground Truth Generation

The Shanghai Tech Dataset [3] is used which has 1198 images. The use of Geometric adaptive kernel as proposed in [1] helps deal with highly congested environments. The dataset is availed with ground truth information computed as discussed in [1]. The geometric adaptive kernel is given in equation 3 below,

$$F(y) = \sum_{i=0}^{M}\delta(y - y_i) * G_{\sigma_i}(Y)\text{ with } \sigma_i = \beta * d_i \dots\dots\dots(3)$$

where *yi* is the object that has been targeted in the ground truth δ. di indicates the average distance of k nearest neighbours. To generate the density map, image is convoluted *δ(y − yi)* with the Gaussian kernel, '*' represents the convolution operator. *y* is the position of pixel in the image, The value of *k* is kept as 3 and *β* as 0.3 as per [1]. For less dense crowds, the Gaussian kernel is adapted to blur the average head size around the positions of heads annotated.

3. Experiments and results

This section demonstrates some of the outputs of the algorithm on the ShanghaiTech testing set of images where the actual count is also available. Some of the results of images are crawled from the internet and demonstrate the results on those images as well. To evaluate accuracy of these images, manually counting is done; hence the actual count also may not be completely accurate due to human error. Validation of those estimated results with the output of the crowd-count algorithm is required.

A. Metrics For Evaluating the Neural Net

The Neural Network is evaluated on the basis of 2 metrics MAE and MSE. Mean Absolute Error (MAE) is absolute difference of 2 continuous variables given below,

$$MAE = \frac{1}{M}\sum|C_i - C_i^{GT}| \dots\dots\dots\dots\dots\dots.\dots(4)$$

Mean Squared Error(MSE) is the average of the squares of the errors - that is the square of the actual value and the value predicted using deep learning from the network,

$$mse = |\frac{1}{M}\sum|C_i - C_i^{GT}|^2 \dots\dots\dots\dots\dots(5)$$

In the above formula, M is single test sequence images and C_i^{GT} represents the count from ground truth and C_i represents the estimated count by using the proposed methodology. The approximated count or estimated count is evaluated as,

$$C_i = \sum_{l=1}^{L}\sum_{w=1}^{W} z_{l,w} \dots\dots\dots\dots\dots\dots\dots\dots..\dots\dots..(6)$$

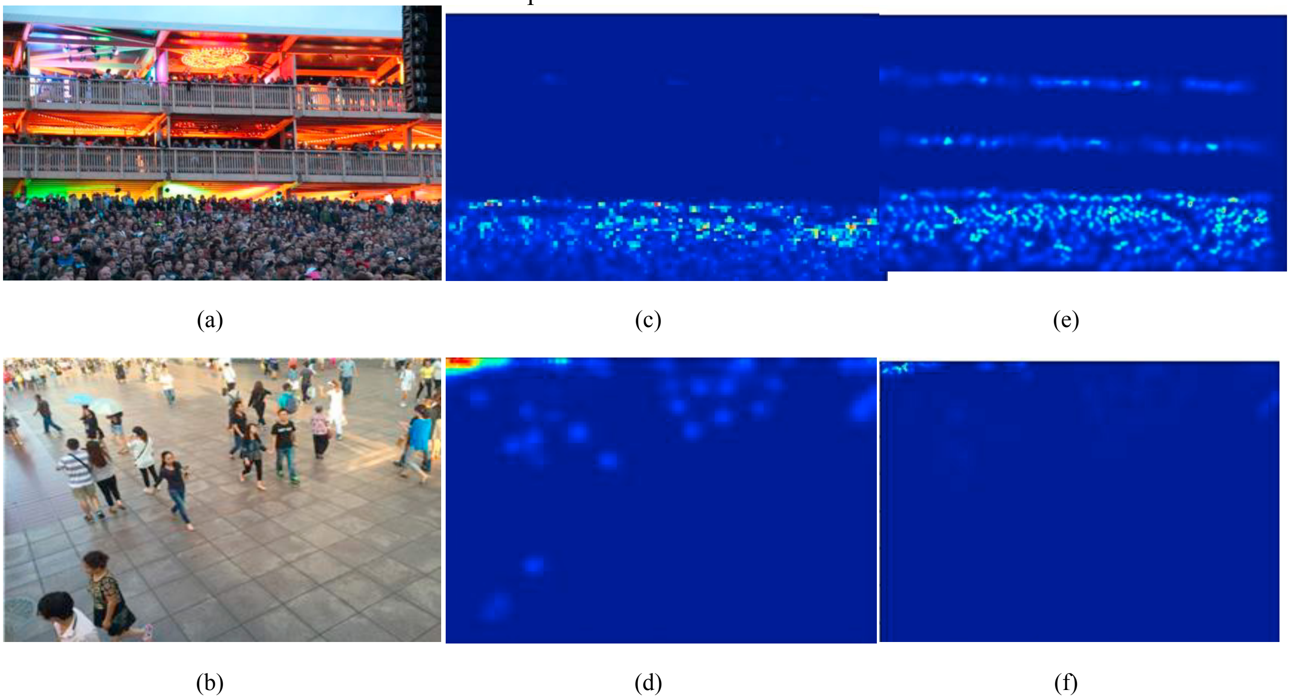
where *L* is the length of density map, *W* is the corresponding width and $z_{l,w}$ is the pixel at location *l,w* generated in the density map.

The ShanghaiTech dataset is one of the most used dataset for crowd-counting applications with 1198 annotated heads. This dataset is available with ground truth.

B. Results on ShanghaiTech dataset

In this framework, target frame is an individual image extracted from video whose crowd count is to be estimated. The input images are fed into the network, sample input images are shown in figure 3 (a) and (b); the network generates a density map as shown in figure 3 (c) and (d) of the sample images. The count can be computed by feeding this density map as a numpy array and computing the addition. The predicted count for sample image shown in figure 3 (a) is 485.123; however ground truth count is 449. Figure 3 (e) and (f) shows the true density map generated by adapting the Gaussian kernel to the ground truth file which contains the location of heads. It is clearly observed from the density map that the network performs very well for dense images. For sparse images, the ShanghaiTech part B model plays a crucial role. In cases of sparse crowds, the visibility of heads of people is clearer instead of using a geometry adaptive kernel for ground truth generation; the Gaussian kernel is initialized to the average human head size.

The MAE for the dense dataset is 67 and for sparse dataset is 10.



(a) (c) (e)

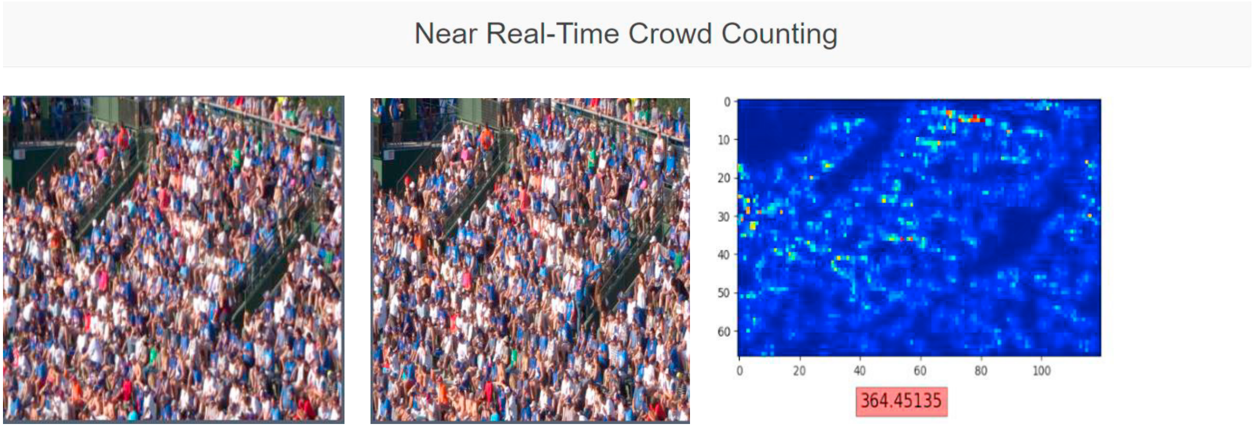
(b) (d) (f)

Figure 3. (a) The Input training image from ShanghaiTech Part A (dense) Dataset sample image, (b) ShanghaiTech part B (sparse) dataset sample image, (c) result generated using proposed approach for ShanghaiTech Part A (dense) Dataset sample image, (d) ShanghaiTech Part B (sparse) Dataset sample image which in fact need not use a geometry adaptive kernel.

The sample input image from figure 3 (b) shows an image with sparse distribution of people. It also shows the versatility of the algorithm to adapt to different environments such as partially visible heads, people using umbrellas, hats etc. The predicted count from the proposed approach is 52.152 whereas the ground truth count is 52.

C. Results on Video Data

The end-to-end system that is designed accepts a video feed as input and it predicts the headcount on the frames created from the video. Since the model requires around 5 seconds to process the frame and predict its headcount, the system is deemed to work in near real-time fashion.



(a) (b) (c)

Figure 4. (a) The video frame whose headcount is to be calculated, which would be fed into the model (b) The video feed of a crowded stadium which is showed throughout to the system user (c) The output of the model with the headcount showing in the bounded box.

The above images show the screen images of the working system. The system shows the continuous video feed at all times in the center panel. The left panel of the screen shows the frame whose headcount is being calculated and the right hand panel of the screen shows the headcount of the frame in a bounded box

D. Performance Details

As the proposed work asserts a near real-time crowd counting system, performance of the network is a key factor for this end-to-end crowd counting application. The model is executed on Google Colab platform using Tesla K80 GPU, and Xeon processor (2.3 Ghz). The training of the network involves 1 epoch with 700 steps per epoch and the batch size kept as 1 for training the model with ShanghaiTech Dataset. The model is also additionally trained on video data with batch size of 16. The parallelization of the network is made possible by use of TensorFlow with GPU accelerated performance on Keras framework. This enabled to train the network in around 2 hours for part A of image dataset of which has dense population and 2 hours for part B images which has sparse population. Training the network on only CPU would have required around 9 hours for part A and another 9 hours for part B of ShanghaiTech dataset. Table 1 show the time taken for entire training process executed using GPU along with CPU and the time for the same configurations running only on CPU. It is observed that the GPU computation performance for training is significant as compared to CPU performance. Another thing that was observed was that the system is affected by the resolution of video. After many trial-and-errors, we have found that our crowd-counting system performs optimally when given a frame of resolution (300*450) pixel. The model takes around 4 seconds to calculate headcount of the inputted frame when its resolution is (300*450) pixel. Large resolution image requires longer time to predict. The performance of the model also depends on the hardware of the system in which the model is run.

Table 1. Only CPU and CPU-GPU computation performance

Particulars	GPU	CPU
Total time taken for training process	4hrs	18hrs

E. Comparison with Other Crowd-Counting Methods

A lot of research has been done in the field of Crowd-Counting. With the improvements in infrastructure and better technologies emerging every day, there has been a significant improvement in the model accuracy. The training times of the models has also decreased with newer infrastructures. The table 2 shows the performance of CSRNet based crowd estimation used in this work as compared to other state-of-the-art crowd-counting models.

Table 2. Accuracy evaluation as compared to other state of art models

Method	**Part_A (Shanghai Tech)**		**Part_B (Shanghai Tech)**	
	MAE	**MSE**	**MAE**	**MSE**
Multi-Column Convolutional Neural Network[1]	110.2	173.2	26.4	41.3
Switching-CNN[31]	90.4	135.0	21.6	33.4
CP-CNN [32]	73.6	106.4	20.1	30.1
SANet[33]	67.0	104.5	8.4	13.6
Image Pyramid[34]	80.6	126.7	10.2	18.3
Spatial-Aware [35]	69.3	96.4	11.1	18.2
CSR Net[29] based model enhanced for stream data	**68.2**	**115.0**	**10.6**	**16.0**

VI. Conclusion and further work

This work presents an approach which will be effective for near real time crowd counting using DCNN. The benefits of the applications includes High Performance Computing through the use NVIDIA GPU parallel framework, a swift and agile method for processing of the video feed taken through a camera with an innovative solutions that can be deployed for disaster management, emergency evacuation without having to configure explicit systems for the same. The proposed system performs admirably in situations where manual counting is simply not possible. Deep learning also enables the system to perform in versatile environments and continuously learn from new inputs. The Experimental results reveal that the proposed methodology achieves promising crowd count predictions almost as good as ground truth. Another major advantage of using the end-to-end application is that no external configurations are required for achieving crowd-count except for the video feed of the particular area.

Further, supportive research is required to remove the complexities such as shadows, non-leaving lookalikes involved in crowd counting to improve the accuracy of the system. The project has a very vast scope in the future and can be implemented on satellite footage in future. The project is flexible in terms of expansion and can be expanded to trace or study the movement of the crowds which could be helpful in managing riots, rallies etc. The proposed system architecture can also be used in monitoring real time traffic by creating density maps for cars.

References

[1] Zhang, Yingying, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. (2016) "Single-image crowd counting via multi-column convolutional neural network." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 589-597.

[2] Simonyan, Karen, and Andrew Zisserman. (2014) "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

[3] Boominathan, Lokesh, Srinivas SS Kruthiventi, and R. Venkatesh Babu.(2016), "Crowdnet: A deep convolutional network for dense crowd counting." In Proceedings of the 24th ACM international conference on Multimedia, pp. 640-644. ACM,

[4] Peng, Lei, Lei Wu, Yalan Ye, Fengqi Yu, and Hai Yuan. (2007), "CPN modeling and analysis of HMIPv6." In 2007 IEEE International Conference on Integration Technology, pp. 68-73. IEEE, 2007.

[5] Image datasets, https://cdn-images-1.medium.com,[online] [accessed on December 2018]

[6] Li, Min, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection." In 2008 19th International Conference on Pattern Recognition, pp. 1-4. IEEE, 2008.

[7] Leibe, Bastian, Edgar Seemann, and Bernt Schiele. "Pedestrian detection in crowded scenes." In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 878-885. IEEE, 2005.

[8] Wang, Lu, and Nelson HC Yung. "Crowd counting and segmentation in visual surveillance." In 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 2573-2576. IEEE, 2009.

[9] Gavrila, Dariu M., Jan Giebel, and Stefan Munder. "Vision-based pedestrian detection: The protector system." In Proc. of the IEEE Intelligent Vehicles Symposium, Parma, Italy. 2004.

[10] Mahadevan, Vijay, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. "Anomaly detection in crowded scenes." In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1975-1981. IEEE, 2010.

[11] Tuzel, Oncel, Fatih Porikli, and Peter Meer. "Pedestrian detection via classification on riemannian manifolds." IEEE transactions on pattern analysis and machine intelligence 30, no. 10 (2008): 1713-1727.

[12] Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57, no. 2 (2004): 137-154.

[13] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In international Conference on computer vision & Pattern Recognition (CVPR'05), vol. 1, pp. 886-893. IEEE Computer Society, 2005.

[14] Wu, Bo, and Ramakant Nevatia. "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors." In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 1, pp. 90-97. IEEE, 2005.

[15] Sabzmeydani, Payam, and Greg Mori. "Detecting pedestrians by learning shapelet features." In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. IEEE, 2007.

[16] Felzenszwalb, Pedro F., Ross B. Girshick, David McAllester, and Deva Ramanan. "Object detection with discriminatively trained part-based models." IEEE transactions on pattern analysis and machine intelligence 32, no. 9 (2010): 1627-1645.

[17] Lin, Sheng-Fuu, Jaw-Yeh Chen, and Hung-Xin Chao. "Estimation of number of people in crowded scenes using perspective transformation." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 31, no. 6 (2001): 645-654.

[18] Wu, Bo, and Ram Nevatia. "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors." International Journal of Computer Vision 75, no. 2 (2007): 247-266.

[19] Li, Min, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection." In 2008 19th International Conference on Pattern Recognition, pp. 1-4. IEEE, 2008.

[20] Ge, Weina, and Robert T. Collins. "Crowd detection with a multiview sampler." In European Conference on Computer Vision, pp. 324-337. Springer, Berlin, Heidelberg, 2010.

[21] Chan, Antoni B., Zhang-Sheng John Liang, and Nuno Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking." In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-7. IEEE, 2008.

[22] Ryan, David, Simon Denman, Clinton Fookes, and Sridha Sridharan. "Crowd counting using multiple local features." In 2009 Digital Image Computing: Techniques and Applications, pp. 81-88. IEEE, 2009.

[23] Chan, Antoni B., and Nuno Vasconcelos. "Counting people with low-level features and Bayesian regression." IEEE Transactions on Image Processing 21, no. 4 (2012): 2160-2177.

[24] Idrees, Haroon, Imran Saleemi, Cody Seibert, and Mubarak Shah. "Multi-source multi-scale counting in extremely dense crowd images." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2547-2554. 2013.

[25] Kong, Dan, Douglas Gray, and Hai Tao. "A viewpoint invariant approach for crowd counting." In 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, pp. 1187-1190. IEEE, 2006.

[26] Cho, Siu-Yeung, Tommy WS Chow, and Chi-Tat Leung. "A neural-based crowd estimation by hybrid global learning algorithm." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29, no. 4 (1999): 535-541.

[27] Chen, Ke, Shaogang Gong, Tao Xiang, and Chen Change Loy.(2013) "Cumulative attribute space for age and crowd density estimation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2467-2474.

[28] Chen, Ke, Chen Change Loy, Shaogang Gong, and Tony Xiang. (2012) "Feature mining for localised crowd counting." In BMVC, vol. 1, no. 2, p. 3.

[29] Li, Yuhong, Xiaofan Zhang, and Deming Chen (2018).. "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1091-1100

[30] Kang, D., Ma, Z., & Chan, A. B. (2018). Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking. IEEE Transactions on Circuits and Systems for Video Technology, 29(5), 1408-1422.

[31] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4031–4039, 2017.

[32] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In IEEE International Conference on Computer Vision, pages 1879–1888, 2017.

[33] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750, 2018

[34] Di Kang and Antoni B. Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In British Ma-chine Vision Conference, page 89, 2018.

[35] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. In Proceedings of the International Joint Conference on Artificial Intelligence, pages 849–855, 2018.