



## Congestion Control early warning system using Deep Learning

N. Sandeep <sup>\*</sup>, N.S. Ragul <sup>\*</sup>, P. Nikil Dhas <sup>\*</sup>, V. Vaishnavi <sup>\*,\*</sup>

<sup>\*</sup> Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India.

\* Corresponding Author: [vaishnavi.v@srec.ac.in](mailto:vaishnavi.v@srec.ac.in)

Received: 01-09-2021, Revised: 27-09-2021, Accepted: 01-10-2021, Published: 30-10-2021

**Abstract:** A new approach is proposed to analyze the live crowd and to provide an alert at the time of congestion, over-crowding and sudden gathering of pedestrians in a particular region. This paper proposes a completely software-oriented approach using MATLAB where it uses object detection and object tracking using Faster R-CNN (Region Based Convolutional Neural Network) algorithm where inception model of Google is used as CNN model which is pre-trained. This proposed method gives significant result on proposed dataset and the crowd congestion using Faster R-CNN approach which gives an accuracy of 93.503% at the rate 28 frames per second and the crowd detected video frames are uploaded to cloud storage.

**Keywords:** Congestion, Object Tracking, Faster R-CNN, MATLAB, Accuracy, Cloud Storage

### Introduction

As the human population all over the world increases day by day and adding to that the wide spread of Covid-19 has caused a severe damage to the human life cycle and had cost lot lives and their life routine both financially and mentally. Crowds occur in a variety of situations like concerts, rallies, marathons, stadiums, railway stations, etc. Crowd analysis is essential from the point of view of safety and surveillance, abnormal behavior detection and thereby reducing the chance of a causing a major disaster. Generally, congestion in the crowd can lead to severe problems like a stampede, suffocation and body pain. This congestion is due to increasing people count in a compact area thereby increasing the crowd density in regions and abnormal crowd motion.

Most of the congestion control approaches follow a hardware-oriented approach. Here in this approach we proposed a software-oriented approach, Congestion Control Early Warning System, for congestion control with the help of object detection and object tracking technique. Object detection is performed by algorithm known as the faster R-CNN architecture in which Google inception model is used as a pre-trained CNN model and with the help of proposed object tracking technique the crowd abnormality is analyzed. The proposed congestion control

technique exhibits quite significant results on the proposed dataset. This proposed methodology is very efficient when compared with other object tracking methods.

This paper proposes an approach to produce alert at the period of congestion in crowd or any other abnormality in the crowd using this Faster R-CNN algorithm. And for people tracking centroid based algorithm using Euclidian's formula. According to these functionalities the crowd is monitored to find and alert the abnormality in certain time period.

## Related Work

As gone through research papers that are similar to our approach in crowd management and alert systems most of the approaches where completely of hardware components and no use of any deep learning algorithms to it.

Mr Victor Hugo Roldão Reis, Mr. Silvio Jamil F. Guimarães and Mr. Zenilton Kleber Gonçalves do Patrocínio Jr has proposed and evaluated the adoption of a capsule network-based (CapsNet-based) model rather than the convolutional neural network-based (CNN-based) models which are predominant in crowd counting tasks. The aim is to join the task of generating a high-quality density map from a single image along with producing a more precise estimate of the number of people. CapsNet-based model has a strong capacity of representation and a powerful dynamic routing mechanism that could address the drawback of a limited number of training samples. The replacement of the scalar values of CNN by vectors when using a CapsNet allows learning more discriminative features, which contributes to generating high-quality density maps, and thus a more precise number of individuals in crowd scenes. Experimental results show that our proposal presents competitive results concerning state-of-the-art, but with a 59.2% reduction in the number of parameters.

CNN (convolutional neural network) is a specialized deep learning model for data processing that achieved great success in several computer vision tasks. It has a strong capability to automatically learn high-level representations from input images. And it consists mainly of three types of layers, namely: (i) convolutional layers; (ii) pooling layer; and (iii) fully connected layers. Relevant information of an image through vectors. Compared to traditional neurons, which produce a scalar value, the capsules can also represent information about the direction related to a feature. Its adoption has achieved excellent results in tasks such as recognition of digits and classification of small images. As far as we know, CapsNet has never be used to tackle the crowd counting problem. Thus, in this work, we propose a new use of CapsNets to the task of crowd counting. We hypothesize that the use of CapsNets can be more efficient for producing high-quality density maps from the complex crowd scenes, dealing better to scale variation and representation in stored in vectors rather than scalars values and using a smaller number of parameters when compared to the state-of-the-art.

Mr Yuanyuan Fan and Qingzhong Liang proposed a method of crowd flow detecting based on RFID by analyzing the factors affecting the RFID link state. Firstly, a pedestrian

detection system-based RFID is allocated at the entrance of channel where reading RFID tags will be affected when people pass through. And then, the Link State Indicator (LSI) according to the counts of reading tags is considered to get coverage information, while moving status can be inferred by RSSI detection. With the evaluation in different scenarios, the coverage and moving status of the crowd can be verified. With the rapid development of urbanization, the organization of pedestrian flows in large public space like airports, train stations, shopping malls etc. is a big challenge for public safety in these spaces. Systems with information about current crowd densities and moving status are able to support the control and management process of pedestrian flows and enhance public safety. The detection of the pedestrian flow in popular urban areas, which focuses on the crowd number, can bring advantages in effectively preventing the occurrence of accidents such as pressing stampede. In the last decades, many methods including on-site supervision by manual checking, cameras with image processing techniques and infrared detecting. However, these techniques require special additional hardware components and suffer from too many factors, such as bad weather, shelters, and so on. Furthermore, implementation cost in order to track people in huge areas has to be discussed carefully in certain scenarios.

In addition, the RFID technology which is mainly based on identifying each electronic tag carried by people has been used to find out the pedestrian flow information for many years. In recent years, WIFI tracking and Bluetooth is also devoted to provide a good approximation to crowd densities and pedestrian flows. In order to reduce the impact of environmental factors, such as weather reasons, atmospheric refraction, equipment pollution, an application system for pedestrian flow detection based on RFID technology is proposed in this paper. Although the cost of personal identification in public places which requires everyone to carry embedded RFID tags is too high, we can still gather the traffic information through RFID signal link status with low cost. In this paper, we design a RFID detecting system in public place, and monitor RFID RSSI (Received Signal Strength Indicator) value changes in different scenarios. With the evaluation, we can verify the coverage changes of the crowd.

### Design and Implementation

Figure 1 represents the block diagram of the congestion control early warning system where the video feed is given as input to MATLAB. Then the bounding boxes are generated for each person using Faster R-CNN algorithm in the video frame. And if the number of detected people count goes higher than the threshold value then an alert message will be displayed. After the detection of congestion or abnormal crowd the detected particular video frame will be captured and stored in a real-time cloud storage platform which can be accessed by authorized clients for their future usage.

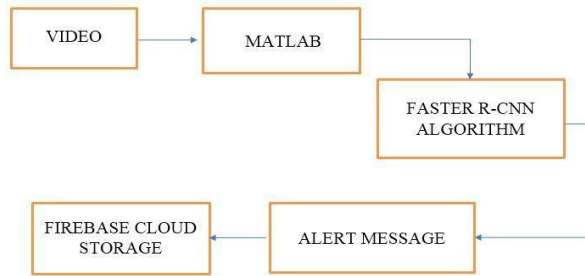


Figure1. Block Diagram

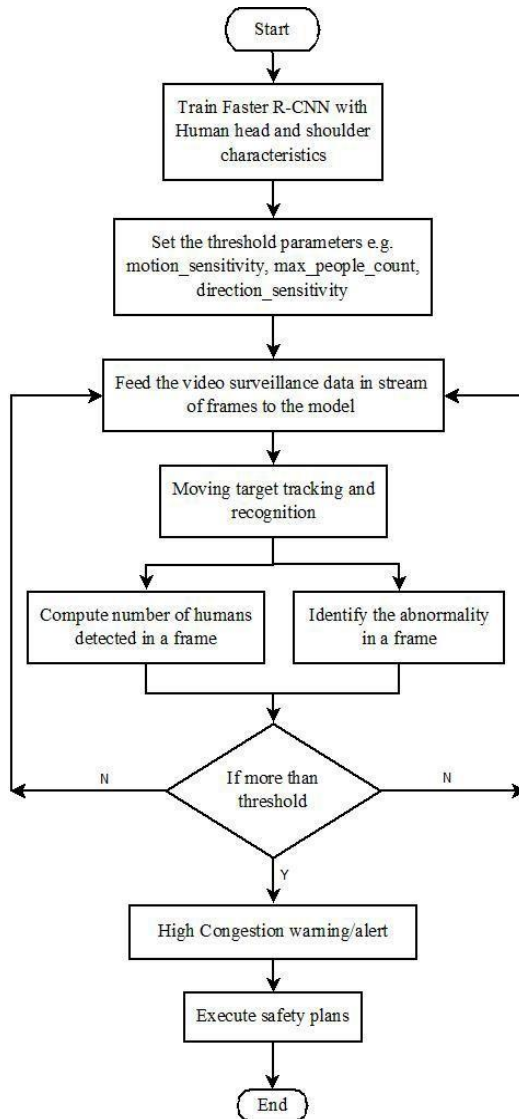
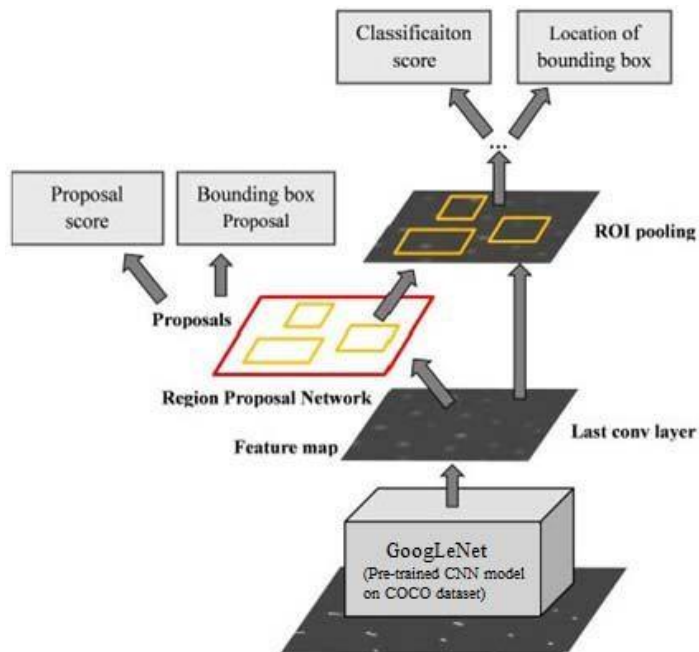


Figure 2. System design

In the proposed system of congestion control on crowd behavior is analyzed with the help of three tasks; object detection, object tracking and object motion direction, where the object is a human. Figure 2 shows the design of the proposed approach for an early-warning system to detect high congestion and generate safety alarms for the execution of safety plans.

### a. Object detection

Faster R-CNN architecture as shown in Figure 3, helps to accomplish the task of object/head detection by deploying the input frame on a pre-trained CNN model, such as Inception architecture of GoogLeNet. Then, Region Proposal Network is used to detect the regions that might contain the objects in the feature map by generating the object proposal score and bounding boxes. Then the Region of Interest pooling layer is used to extract the feature maps according to the regions proposed by RPN and the feature maps output from CNN. Finally, the output feature map is then used for classification and in fine-tuning of the bounding boxes via fully connected Layers.



**Figure3.** Faster R-CNN architecture with GoogLeNet as pre- trained CNN model.

### b. Object Tracking in Crowd

This paper proposes a novel method to track the object. The idea is that the movement of a particular object in the consecutive frame is gradual, i.e. the distance traveled is less. The procedure to track an object is as follows.

- Compute the centroids of the bounding boxes of the objects in the consecutive frames.

- Centroids, which are closer to each other in the consecutive frames, belong to the same object. The Euclidean distance is used to measure closeness between the centroids of the two frames.
- If closeness property is not satisfied, then that object is treated as a novel object.

For instance, consider two detected objects as shown in figure 4 (denoted by black color bounding box) with the tag as ID 1 and ID 2. Suppose they moved at a certain distance and their new position is represented by a red color bounding box. Now compute the distance of the centroid of ID 1 with the new centroids. The centroid which is closer to ID 1 is provided an ID 1 tag. Similarly for ID 2. The new tag is given to the object that has no such closer centroid.

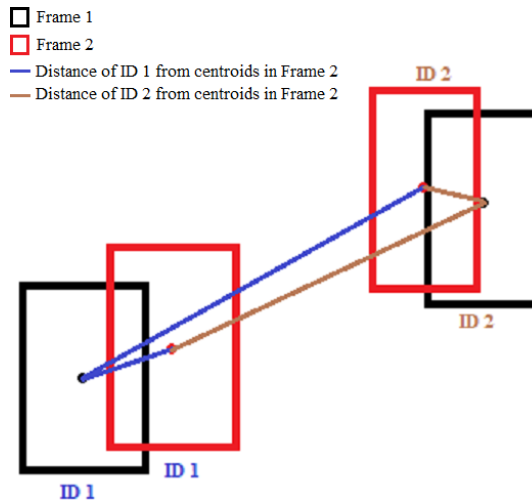


Figure 4. Object tracking approach

## Implementation

The proposed system is implemented using and is visualized with the help of flags concept.

Consider two flags, *good\_flag*: Indicating the normal situation of the crowd and *bad\_flag*: Indicating the abnormal situation of the crowd, leading to high congestion.

1. Train the faster R-CNN model for human head detection on the proposed dataset.
2. Do object detection and get coordinates of bounding boxes.
3. Find the area and centroid for each detected bounding box.
4. Maintain the previous areas and previous centroids to compare the boxes in the consecutive frames.

5. Track the objects by using proposed centroid based object tracking algorithm.
6. Compare area of bounding boxes in two frames belonging to the same object to get the direction of motion of the crowd.
7. If the total number of people is greater than the number of people allowed in the frame
  - If the motion is in one direction (either towards the camera or away from the camera) then increment the good\_flag by 1. (Increment is done by two because of this being excellent condition than abnormal).
  - If an equal number of people are moving in the opposite directions to each other, i.e. both towards and away from the camera then increment the bad\_flag by 2. (Increment is done by two because of this being terrible condition than usual)
  - If an unequal number of people are moving in the opposite direction to each other, i.e. both towards and away from the camera then increment the bad\_flag by 1.
  - If an object/person stays still for some time (computed by tracking the number of frames in which the person stays still) then also increment the bad\_flag by 1.
  - This condition also follows the sensitivity parameter which ranges between 0 and 1; 1 being more sensitive. 1 means that if people are moving in one direction, then no one can come in the opposite direction, else increment the bad\_flag by 1.
8. If in a frame the total number of people is less than or equal to the number of people allowed.
  - Then increment the good\_flag by 2 (this is the most relaxed condition).
9. Do steps 5, 6, 7, 8, 9 for 20 frames. (Number of frames equal to 20 is not fixed can be changed as per requirement, but for accuracy concerns, it should be 10 to 50).
10. Finally, if bad\_flag > good\_flag then generate the high congestion alert and store the particular frame in the output for analysis and execution of contingency plans.

## Methodology

Figure 5 gives the view of the dataset images used to train the algorithm we have implemented. This proposed algorithm is trained with 4,372 dataset images to increase the efficacy of the algorithm and to decrease the time delay for object tracking and detecting. The congestion scenario is created by making motion as random as possible and label the corresponding node abnormal.

Faster R-CNN model is trained and tested with the shot video frames. These video frames are manually annotated by the bounding boxes around the human head. The model is trained to learn the characteristic features of the human head even with the different orientations.

The model was trained until there was consistently low loss, thereby it almost took 22,000 iterations for training as shown in figure 6.



Figure 5. Trained Dataset

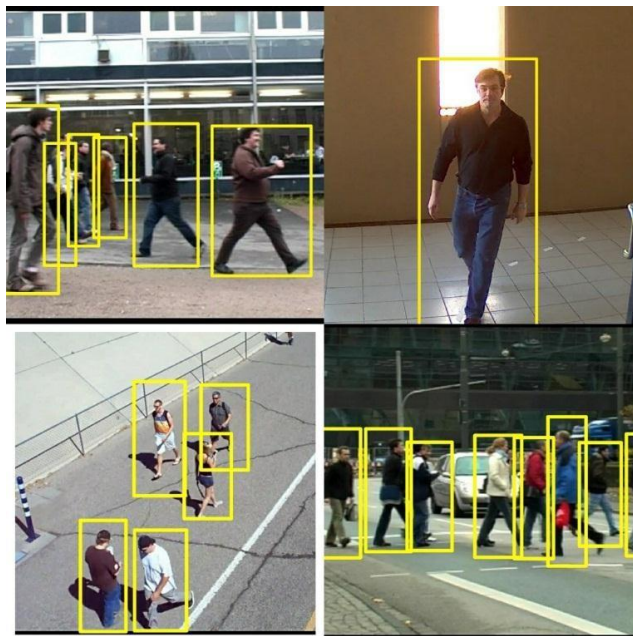


Figure 6. Training Bounding Box

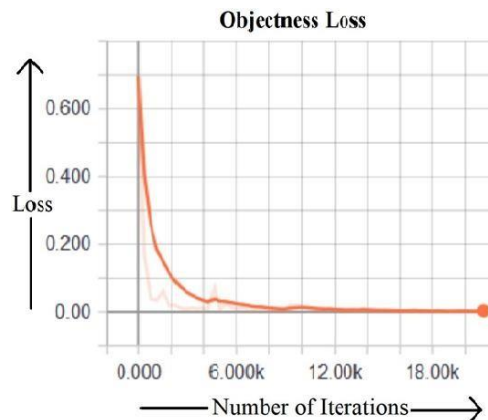
Faster R-CNN a deep convolutional neural network (output from RPN) and fast RCNN detector (final output). Neural networks are trained using stochastic gradient descent and require



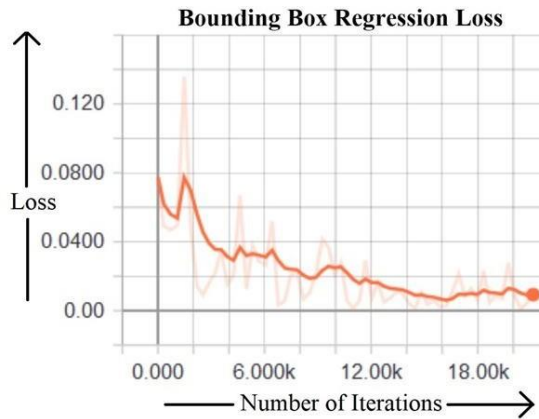
that you choose a loss function when designing and configuring your model. There are many loss functions to choose from and it can be challenging to know what to choose, or even what a loss function is and the role it plays when training a neural network. The role of loss and loss functions in training deep learning neural networks and how to choose the right loss function for your predictive modeling problems. A deep learning neural network learns to map a set of inputs to a set of outputs from training data. We cannot calculate the perfect weights for a neural network; there are too many unknowns. Instead, the problem of learning is cast as a search or optimization problem and an algorithm is used to navigate the space of possible sets of weights the model may use in order to make good or good enough predictions. Typically, a neural network model is trained using the stochastic gradient descent optimization algorithm and weights are updated using the back propagation of error algorithm. The “gradient” in gradient descent refers to an error gradient. The model with a given set of weights is used to make predictions and the error for those predictions is calculated. The gradient descent algorithm seeks to change the weights so that the next evaluation reduces the error, meaning the optimization algorithm is navigating down the gradient (or slope) of error.

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

**RPN Losses:** Binary classification loss or objectness loss (based on object proposal score for being background or foreground) is shown in figure 7 (value: 8.2006e-3), and bounding box regression loss (based on bounding box coordinates proposal) is shown in figure 7 and 8 (value: 0.01053). Smoothing value is just for visualization of the graph, i.e. to make the curve as general as possible; not utilized in any computation. In the following graphs the dark orange colored curve is the generalized representation of the actual curve indicated by light orange color.



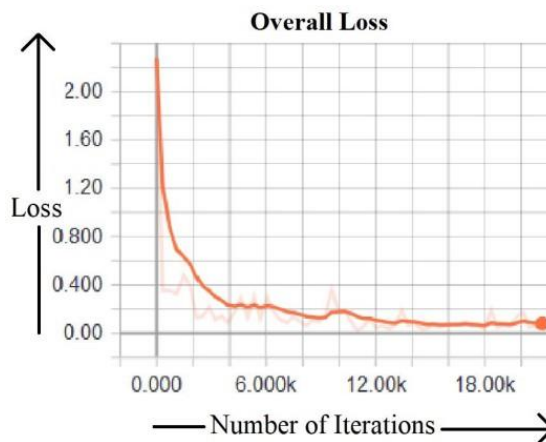
**Figure 7.** Binary Classification loss over the number of iterations (smoothing value: 0.8)



**Figure 8.** Bounding Box regression loss over the number of iterations (smoothing value: 0.8)

Classification Losses: Classification loss (based on some class labels) and bounding box localization loss (based on the fine-tuned bounding box coordinates).

Since only one type of class label was needed (head) and the objective of the second module of faster R-CNN is to classify the detected objects into multiple classes; thereby only the first module of faster R-CNN is needed to serve the purpose of human head detection. Thus, the overall loss (value: 0.06497) is computed as the weighted sum of the classification and the regression loss as shown in figure 9.



**Figure 9.** Overall loss over the number of iterations (smoothing value: 0.8)

The values are set pertaining to the experimental environment. These values are not fixed and may vary for a different environment. Table 1 shows the threshold parameters and respective values, which were kept during experimentation.

### Experimental Results and Discussion

Bounding box is generated for each and every person in the video frame. And each person in the video frame is numbered uniquely until they leave the video frame and the number of people count starts increasing as the number of people enter into the video frame.

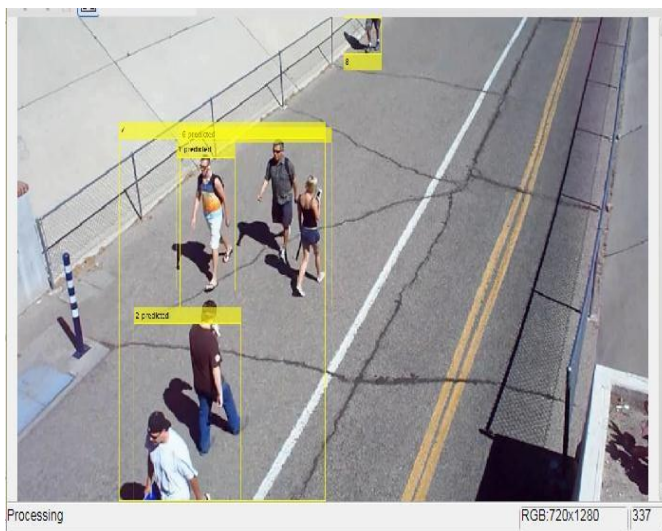


Figure10. Generation of Bounding Box

When more than 30 people and two congestion is detected in the video frame , which is more than the threshold value (30) set for the people count and the congestion count (1) the warning dialog box will be popped out saying “More people and congestion”. And the command window displays the total people count and the congestion count.



Figure 11. Crowd Alert

When the total number of people in frame is very less than the threshold value until the flow of input video frame then the popup window displays as “Less Crowd”. This informs the guard or the surveillance official that no crowd is detected until the given input video ends. The command window displays the total count of people in the video frame and compares it with the threshold value and pops the message window.



Figure 12. Less Crowd Detection

The crowd detected video frames are sent to firebase cloud storage for the future use, if there is any need to know the count of crowd in a particular day. So the implementation of cloud storage would be more favorable for the user to fetch the data at any time and any place where he is in need of it.

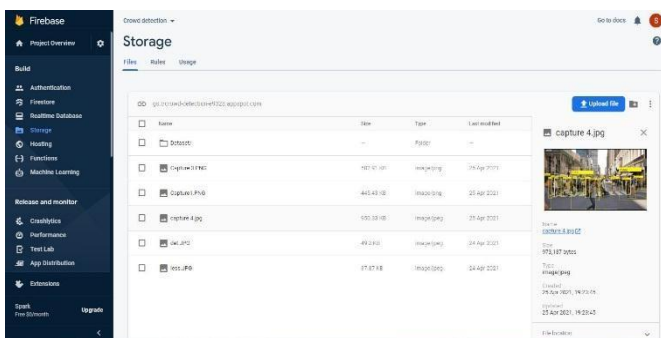


Figure 13. Cloud Storage

Accuracy is the important parameter that has to be analyzed to find the efficiency and the performance of the proposed technique to solve the problem. In this project we have

compared the proposed algorithm with other two similar algorithms to find the most efficient and user friendly way to solve the raised issue.

The faster R-CNN model along-with google inception resnet v2 CNN model produced significant results by detecting the humans in the respective frames with an accuracy of 93.503% at the rate of 28 FPS. The detection accuracy and the output FPS comparison with the other state-of-the-art methods like R-FCN and SSD tested on the same proposed dataset and environment. The overall human detection accuracy and output FPS for Faster R-CNN is significantly better than other methods.

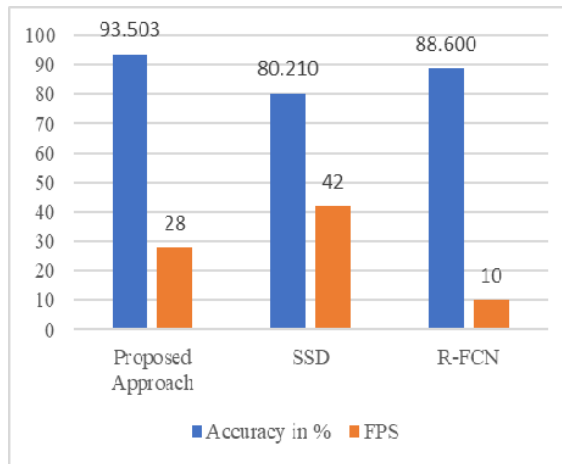


Figure 14. Human detection accuracy and output FPS comparison

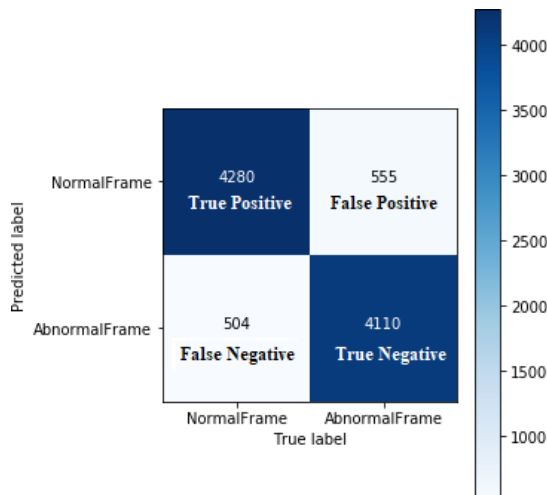


Figure 15. Confusion matrix between true label and predicted label for normal and abnormal frame

The performance of the proposed approach can be observed from the following confusion matrix. The overall accuracy, precision and recall for the proposed method is 88.79 %, 88.52% and 89.46%.

## Conclusion and Future Scope

This paper proposes the novel Faster R-CNN architecture to analyze the crowd behavior and generate the warning/alert signals to execute the contingency plans in order to control the congestion and prevent from mishap. The system follows three tasks of object detection, object tracking, and object motion direction. Each of this task is achieved by following the modified faster R-CNN architecture (because of the requirement of one class classification only first module of faster R-CNN, RPN is required), centroid-based algorithm and analysis of the output of these two tasks for abnormality detection concerning the crowd motion respectively. And through testing, it was found that alert signals were generated for every corresponding frame for which there was a higher chance of occurrence of mishap like stampede due to congestion in crowd motion. And the crowd detected video frames are stored in cloud for further accessibility of user.

Based on the accidents and the wide spread of Covid -19 that can happen due to abnormal crowd behavior, there is a huge need for research in the area of crowd behavior management. The proposed approach is efficient enough to execute the contingency plans for the detected crowd congestion, but still, it can be improved further. The one challenging task which can be tackled is of occlusion; thereby this approach can be enhanced further to handle the cases where people are hidden behind some other person or object as per camera vision. The Frames per Second (FPS) of the output can be increased by following different and new architectures. The approach can be updated even to recognize people in the low-resolution frames in which the head is barely visible. Indeed, there are countless possibilities for improvements in the field of crowd behavior analysis. With the passage of time technology is evolving and new types of challenges are originating which are leading towards new corresponding research fields.

## References

- [1] Victor Hugo Rold, Reis, Silvio Jamil F. Guimar, Zenilton Kleber Goncalves do Patroc, (2020). Dense Crowd Counting with Capsule Networks, *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, Brazil. <https://doi.org/10.1109/IWSSIP48289.2020.9145163>
- [2] Pradeepa B., Viji A., Joshan Athanesious J., Vaidehi V., (2020). Anomaly Detection in Crowd Scenes using Streak Flow Analysis, *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, IEEE, India. <https://doi.org/10.1109/WiSPNET45539.2019.9032845>

- [3] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto & Nicu Sebe, (2018). Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection, *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, USA. <https://doi.org/10.1109/WACV.2018.00188>
- [4] Yuanyuan Fan & Qingzhong Liang, (2017). An Improved Method for Detection of The Pedestrian Flow Based on RFID, *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, IEEE, China. <https://doi.org/10.1109/CSE-EUC.2017.23>
- [5] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, Kevin Murphy, (2007). Speed/accuracy trade-offs for modern convolutional object detectors, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, USA. <https://doi.org/10.1109/CVPR.2017.351>
- [6] Shangnan Liu, Qiang Cheng, Zhenjiang Zhu, & Hao Zhang, (2016). Analysis and Design of Public Places Crowd Stampede Early- Warning Simulating System, *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, China. <https://doi.org/10.1109/ICIICII.2016.0058>
- [7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, (2016). Ssd: Single shot multibox detector, *European conference on computer vision*, Springer, 21-31. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- [8] Pathan S., Al-Hamadi A., & Michaelis B., (2010). Crowd behavior detection by statistical modeling of motion patterns, *2010 International Conference of Soft Computing and Pattern Recognition*, IEEE, France. <https://doi.org/10.1109/SOCPAR.2010.5686403>
- [9] N. Farooqi, Intelligent safety management system for crowds using sensors, *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, IEEE, Cambridge. <https://doi.org/10.23919/ICITST.2017.8356365>
- [10] Vidyasagaran S., Devi S.R., Varma A., Rajesh A., Charan H., (2017). A low cost IoT based crowd management system for public transport, *2017 International Conference on Inventive Computing and Informatics (ICICI)*, IEEE, India. <https://doi.org/10.1109/ICICI.2017.8365342>
- [11] Rohit K., Mistree K., Lavji J., (2017). A review on abnormal crowd behavior detection, *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, IEEE, India. <https://doi.org/10.1109/ICIIECS.2017.8275999>
- [12] [https://en.wikipedia.org/wiki/2017\\_Mumbai\\_stampede](https://en.wikipedia.org/wiki/2017_Mumbai_stampede), as on 29th Oct., 2018.

### **Funding**

No funding was received for conducting this study

### **Conflict of interest**

The Authors have no conflicts of interest to declare that they are relevant to the content of this article

### **About The License**

© The Author(s) 2021. The text of this article is open access and licensed under a Creative Commons Attribution 4.0 International License