

Crowd Size Estimation and Detecting Social Distancing using Raspberry Pi and OpenCV

Davidson Kamala Dhas Milton, and Arun Raj Velraj

Abstract—In this covid19 pandemic the number of people gathering at public places and festivals are restricted and maintaining social distancing is practiced throughout the world. Managing the crowd is always a challenging task. It requires monitoring technology. In this paper, we develop a device that detects and provide human count and detects people who are not maintaining social distancing . The work depicted above was finished using a Raspberry Pi 3 board with OpenCV-Python. This method can effectively manage crowds.

Keywords— Haar features; human detection; distance checking; raspberry pi; opencv

I. INTRODUCTION

THE Internet of Things (IoT) suggests for real time applications that are using sensors, programming, and advancements. It also speaks with other devices over the internet networks. The procedure of observing and controlling colossal social affairs of individuals for their wellbeing and security is known as group checking. In this pandemic condition, public safety has become extremely difficult to monitor. How much individuals permitted to gather in broad daylight spaces, for example, roads, train stations, and celebrations is restricted, and keeping social separation is empowered. So, in those conditions observing the group utilizing cameras will be gainful. We can monitor the people and calculate the distance between them in pixels. Managing the crowd entails identifying each person in the camera's feed and delivering the total number of persons present. It also entails measuring the distance between each other and detecting if the distance is smaller than the distance already provided. The general technique for object detection is background elimination under the situation of fixed cameras. Inter-object occlusion is a common problem in real-time detection, making it a difficult task in computer vision. There have been examinations for human identification considering foundation demonstrating, head shoulder location, human head discovery, etc.

This study centers around separating the caught video into outlines and Haar overflow is utilized to handle the video approaches that identifies people in that specific edge and assuming they are nearer than the doled out distance esteem it will be shown with some sort of caution.

The paper is fanned out as follows that connected works in this field are remembered for Section II. Segment III depicts the proposed work; Section IV frameworks the work's equipment execution; Section V portrays the strategy utilized in the work; The results are point by point in Section VI, and the end is in

Section VII. The work's references are kept in Reference section.

II. RELATED WORKS

Many works have been done to keep a track of crowd counting and maintaining social distancing among the public.

The idea for "Realization of Multiple Human Head Detection and Direction Movement Using Raspberry Pi" came from Syed Ameer Abbas, M. Anitha, and X. Vinitha Jaini [1]. A Raspberry Pi 3 board is used to build a system that uses OpenCV-Python to recognize human heads and provide a count of the people in the area. This study aims to develop a cascade classifier for identifying human heads using samples. After that, the video frames in which the human heads are identified and a count of the people in the scene are processed by the trained cascade. Training the cascade for head detection enables effective detection in locations with inter-object occlusion. After that, the optical flow method is used to follow the human heads that have been observed. The scene's people's motion direction is determined by this tracking.

"Monitoring Social Distancing for Covid- 19 Using OpenCV and Deep Learning" was proposed by Rucha Visal, Atharva Theurkar, and Bhairavi Shukla [2]. Closed-circuit television (CCTV) and drones are used in this implementation, with the camera using object detection to determine the crowd's distance from one another. Pixels are used to calculate and compare the Euclidean distance between two people to a particular standard distance; The local authorities or local police authorities are informed if the Euclidean distance is found to be less than the standard distance.

"Real Time Crowd Counting using OpenCV" was proposed by Shrushti Rohidas Mhaske, Sonal Rajesh Yeshwantrao, Ayush Kumar [3]. A standard CPU can be used to run this system at the same time. To improve human detection accuracy, deep learning object detectors are utilized. For improved tracking accuracy, it also makes use of Centroid tracking and correlation filters, two distinct object tracking methods that enable it to both recover individuals who may have become "stuck" during the process of tracking and discover new individuals. This method can also be used to count the number of vehicles on the road.

"People counting system using raspberry Pi with opencv" was proposed by Badhan Hemangi and K. Nikhita [4]. A method for using a fixed, inexpensive Picamera mounted vertically on a Raspberry Pi board and a Python programming tool attached to the application to count people passing through a virtual gate is

Davidson Kamala Dhas, Associate Professors/ECE at Mepco Schlenk Engineering College (Autonomous), Sivakasi, TamilNadu, India (e-mail: mdavid@mepcoen.ac.in).

Arun Raj, Assistant Professors/ECE at Mepco Schlenk Engineering College (Autonomous), Sivakasi, TamilNadu, India (e-mail: arunraj@mepcoen.ac.in).



described in this paper. The findings indicate that because it is more accurate, counting people with a camera is superior to using other sensors for large entrances. However, it also demonstrates that significant changes are required for the system to be truly trustworthy.

"Literature Survey of Iot Capabled Crowd Analysis Using Raspberry Pi-3" was proposed by Kanchan Mangrule, H. T. Ingale, Vijay D. Chaudhari, and Dr. A. J. Patil [5]. Using Open CV-Python and a Raspberry Pi 3 board with an ARMv8 CPU, a system that recognizes human heads and provides a count of individuals in the area is developed here. Oxygen levels can also be monitored with sensors. The orientation of the crowd and whether it is harmful are both determined by this model. This study examines a variety of methods, including crowd density estimation, tracking, motion detection, and behavior identification, for determining abnormal crowd behavior. They have detected not only crowd behavior but also weapons like guns and other weapons of this kind for the purpose of increasing security. Additionally, it stated which side the weapon is on. According to this method, deep learning is the most fascinating and effective machine learning technology now.

Md Israfil Ansari , Shim Jaechang [6] proposed "People Counting System using Raspberry Pi". Based on blob recognition and blob tracking, this research provides a low-cost approach for counting individuals. Background subtraction is utilised to detect the blob, which is then categorised based on its width and height to determine whether the blob is a person. In this approach, we first determine the video frame's entry and exit points. When the middle of the people blob crosses the designated point, the counting of people begins. Finally, the total number of persons who entered and exited the location is shown. The proposed system's experiment results show that it has a high level of accuracy in real-time performance.

"Crowd Video Count in Low Resolution Surveillance Head Detector and Color based • using Segmentation for Disaster Management" was proposed by Jaysri Thangam, Padmini Thupalli Siva, and B. Yogameena [7]. In this instance, a general head detector as well as a color-based segmentation algorithm were utilized. To begin, individuals are identified using color-based segmentation to determine their skin tone. After that, the scene will be divided into four quadrants, and a standard head detector will be used to count the number of people in each quadrant. Squint-viewed, medium-density crowds with heavy occlusion are the scope of this project.

Songyan Ma , Tiancang Du [8] proposed "Improved Adaboost Face Detection". The color segmentation is done using statistical properties of facial skin color, and then the trained Adaboost algorithm classifier cascade is utilized to create face detection and gain precise positioning of the human face.

"Estimating the Number of People in Crowded Scenes by MID Based Foreground Segmentation and Head-shoulder Detection" was proposed by Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan [9]. A head-shoulder recognition algorithm based on HOG (Histograms of Oriented Gradients) and a foreground segmentation algorithm based on MID (Mosaic Image Difference) are combined in this proposed approach to provide an accurate count of people in the observed area. For the head-shoulder detection module to count people

and identify their heads, the MID-based foreground segmentation module of our system creates active zones.

"Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera" was written by Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and LucVan Gool [10]. In this study, a monocular, possibly moving, uncalibrated camera is used to automatically detect and track a variable number of people in difficult environments. In a particle filtering framework, a novel approach to multiperson tracking-by-detection is presented. This method uses the continuous confidence of pedestrian detectors and online-trained, instance-specific classifiers as a graded observation model in addition to final high-confidence detections. As a result, instance-specific data are added to knowledge of general object categories. The main idea of this paper is to look into how these bad sources of data can be used to track multiple people reliably. Without the use of backdrop mode, camera or ground plane calibration, or historical data, the system detects and tracks a large number of people who are constantly moving in complex environments with occlusions. Consequently, it is suitable for web applications and has few limitations. In a wide range of extremely dynamic scenarios, such as traditional surveillance records, webcam footage, or sports sequences, our findings demonstrate that the method produces good tracking performance. As shown, this method performs better than other algorithms that use more data.

"Tracking Multiple Humans in Complex Situations" was proposed by Tao Zhao and Ram Nevatia [11]. Using ellipsoid human form models, multiple human objects are segmented and their global motions are tracked in 3D. It works in situations where a small group of people walk together, there is occlusion, and there is cast shadow or reflection, according to experiments. Locomotor modes (such as walking, running, and standing) and 3D body postures are inferred from a previous locomotion model. Geometric constraints are imposed on both sections by the camera model and ground plane assumptions. On some difficult sequences, solid results are displayed.

"Social Distancing Analyzer Using Computer Vision and Deep Learning" was proposed by G V Shalini, M Kavitha Margret, M J Sufiya Niraimathi, and S Subashree [12]. The development of a method for analyzing social distance is suggested. The system makes use of computer vision and a deep learning model. Computer vision can easily estimate the distance between each person. If a group of people is found to be exceeding the minimum acceptable threshold value, a red bounding box will be displayed. The developed system uses a pre-filmed video of people on a crowded street. The proposed system can determine the distance between two people. The distance patterns of social separation are classified as "safe" and "unsafe." Based on the object detection and classification, labels are also displayed. Real-time applications and live video feeds can both benefit from the classifier.

The process of determining whether or not there are any human beings present in an image is known as human detection. This is accomplished by comparing a small portion of each location in the image to previously established templates or patterns of people. For abnormal event detection, human gait characterization, congestion analysis, person identification, gender classification, and elderly fall detection, a video surveillance system must accurately recognize people. Identifying a moving object is the first step in the detecting

process. Background subtraction, optical flow, and spatiotemporal filtering are all options for object detection. After being spotted, information based on shape, texture, and motion can be used to classify a moving object as a human. In addition to providing comparisons, this study provides a comprehensive evaluation of the various methods that are currently available for recognizing people in surveillance footage.

Both machine learning and deep learning methods can be used to find objects. The machine learning strategy calls for a variety of ways to define features, and then any method, like Support Vector Machines (SVMs), can be used to classify them. On the other hand, the deep learning method makes it possible to finish the entire detection process without explicitly identifying the features that will be used to classify the data. Convolutional neural networks, or CNNs, play a significant role in the deep learning approach. Region Proposals (R-CNN), You Only Look Once (YOLO), deformable convolutional networks, refinement neural networks for object detection (RefineDet), and retina-net are among the various deep learning models.

III. PROPOSED WORK

The proposed work is to foster a framework for human recognition and then some and furthermore checking the distance between everyone utilizing Raspberry Pi 3 board and openCV. OpenCV-Python is used in visual studio code platform which performs the detection process. At first Haar highlights are utilized to prepare a specific picture/input video/constant catching video. Everyone is recognized, and a rectangular box is presented for them. The innovation utilizes a camera to catch the jam-packed scene. The Raspberry Pi 3 contains a quad center ARMv8 focal processor unit that processes video outline by edge to identify the presence of people utilizing a formerly shown overflow classifier. This procedure picks how much people in the get-together. The system is given a desired distance value, and during real-time detection, the distance between one person and another is measured and compared to the desired distance; If the distance is smaller than the required one, the person will be identified by a red rectangular box, while others will be identified by a green rectangle box. Figure 2 depicts the flow of this project.

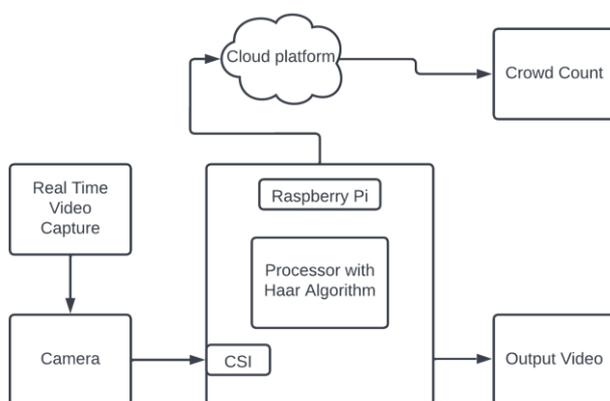


Fig. 1. Crowd Detection and monitoring Block Diagram

Fig1 illustrates the block diagram of crowd detection and monitoring through raspberrypi. A real-time video is captured using a camera linked to the Raspberry Pi. The collected video is then fed into the opencv-python code, which is executed using the visual studio coding platform. The whole human body is detected from the input footage using the Haar cascade technique utilized in the application. After processing the video frame by frame the output video in which persons are discovered and the video in the monitor to which Raspberry Pi is connected displays the social distancing that has been observed. Then the number of people and the number of people violating social distancing are stored in Thingspeak- an IOT cloud platform.

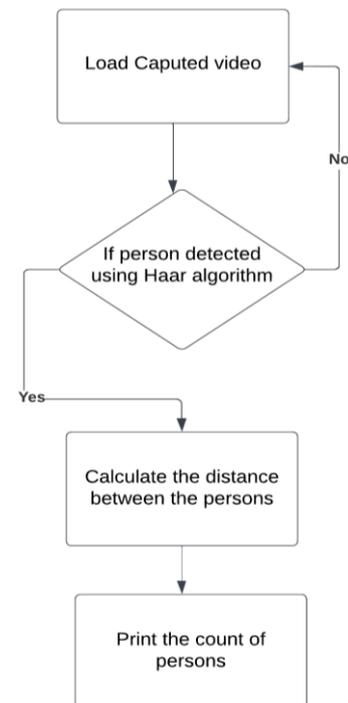


Fig. 2. Flow chart to predict Crowd count

Fig 2 depicts the flow of the work. Initially, a video input is acquired using a camera, or a video can be delivered as an input directly. People in the video will be recognized using the cascade method and shown in a rectangular box. The crowd size is estimated by uploading a count of people to the cloud, which can then be analyzed. The distance between one person and the other is then calculated using pixel values in consecutive frames to see if people are maintaining social distance. The measured distance is then compared to the minimum distance that has already been specified. People will be depicted with a red rectangle box if the distance is less than the minimum distance, or with a green rectangular box if the distance is greater. The number of persons who break social distancing rules is also kept in the cloud, and a warning or alert message can be sent to the authorities.

IV. HARDWARE AND SOFTWARE TOOLS

The Raspberry Pi 3 board with camera point of affiliation is displayed in Figure 3. OpenCV-Python was utilized in this project. OpenCV is a programming functions library. It's mostly for real-time computer vision++, Python, Java, and other

programming languages are supported by OpenCV. A statistical machine learning library is included in OpenCV to help with a range of applications.

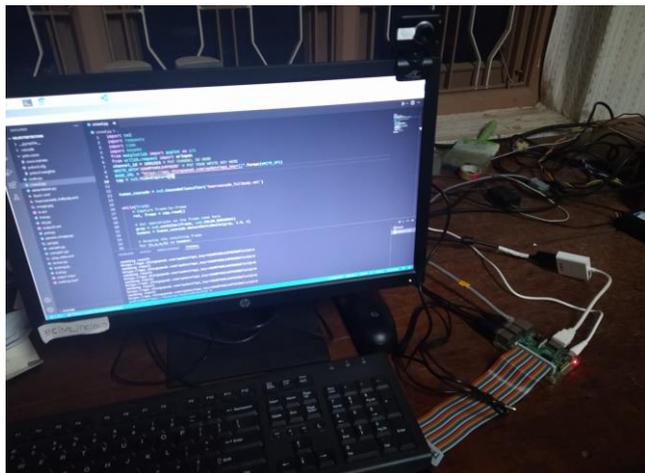


Fig. 3. Implementation of Algorithm in RPi

TABLE I
RASPBerry PI 3

Components	Features
Processor	1.2GHz quad-core ARMv8
Memory	1GB
USB ports	4
GPIO pins	40
Other interfaces	Ethernet port, Camera ,Display and HDMI port

A. Training

To detect human heads in a crowd, the initial step is to train a Haar cascade classifier. For cascade classifier training, positive and negative samples are gathered. The positive samples (foreground images) are photographs with humans, while the negative samples (background images) are images without humans. Separate videos were captured in a human-populated and non-human-populate location. Following that, the videos are processed, and the frames are divided. After that, the cascade classifier is trained to create a series of weak classifier stages. The classifier uses the Haar properties acquired from the positive samples. A cascade classifier is made up of a sequence of weak classifiers, each which classifies the sub-window based on its own set of attributes. The trained cascade is saved as an .xml (Extensible Markup Language) file.

B. Detection

Each frame of the video is read frame by frame, with each frame passing through all stages of the trained cascade classifier. Several sub windows examine each frame. The current stage of the cascade either passes or refuses the examined window to the next stage. A window must complete all the stages. At any point during the classification process, it could be rejected. A window is identified as human after it has been through all the stages.

C. Distance Measurement

Pixel values between people are calculated and compared with the values that are already provided to the system. This indicates whether the distance between the people is as expected.

D. Haar Cascade Algorithm

Haar Cascade is a machine learning-based method that involves using a large number of positive and negative images to train the classifier. Image features such as points, edges, or patterns are used to identify an object in an image. A cascade classifier uses these visual cues as features to determine if an object is in the image, such as a face.

The features on the image make it simple to locate the images edges or lines, as well as locations where the intensities of the pixels abruptly shift. The goal is to calculate the sum of all image pixels in the Haar feature's darker part, as well as the sum of all image pixels in the Haar feature's lighter part. Then figure out how they differ. Edges in opposite directions, as well as any other image structures, will be detected by other Haar properties. To detect an edge anywhere in the image, the Haar feature must traverse the entire image. One of the advantages of the Haar cascade is that it uses integral images to speed up the computation of Haar-like features. They are also extremely efficient at feature selection when using the AdaBoost algorithm.

E. Computer Vision for Human Detection

Computer vision is a technique for manipulating and retrieving data from movies and photos as well as understanding how they are stored. Computer vision is the foundation of artificial intelligence or a major component of it. Computer vision is used extensively in robotics, self-driving cars, and photo editing software. OpenCV is a significant open-source library for computer vision, machine learning, and image processing that is currently used a lot in real-time operations. It can be used to find things, faces, and even human handwriting in pictures and movies. When paired with other modules like numpy, Python can process the OpenCV array structure for analysis. To identify visual patterns and the various characteristics they possess, we employ vector space and perform mathematical operations on these features. OpenCV's features include image/video I/O, processing, and visual display; object and feature detection; monocular or stereo computer vision based on geometry; computer-assisted photography; clustering; and machine learning. OpenCV's HOG implementation is a highly fast human detection algorithm (Histograms of Oriented Gradients). This method has been taught to recognize pedestrians, who are typically visible people who are standing up.

V. METHODOLOGY

A. Video Streaming

For object identification, the scenes are caught by a camera and the video is given as contribution to openCV and compute the Frames Per Second (FPS) throughput rate.

B. Preprocessing

Frames are pre-processed. OpenCV is a package that allows to do typical computer tasks activities involving vision and image processing. All of the output frames will be handled by openCV.

C. Object Detection

A type of PC innovation observes examples of semantic components in pictures and films and draws bounding boxes

around them. It necessities to oversee PC vision and picture taking care of.

D. Object to Object gap

Pixel values between people are calculated and compared with the values that are already provided to the system. This determines whether the distance between people is exactly as expected.

VI. RESULTS AND DISCUSSION

The information is given as a pre-kept video of individuals in a jam-packed climate. The Raspberry Pi is utilized to perceive individuals in the video, which is shown by a blue rectangular box in Fig 4.



Fig. 4. Detection of People in crowd

In addition, as illustrated in fig 5, the number of persons is stored in ThingSpeak as a graphical and numeric representation for both date-wise and timewise.



Fig. 5. People Count in online web portal

People who avoid others are addressed by red boxes, while the individuals who avoid others are addressed by green boxes, as represented in Figure 6 and the number of people violating social distancing is stored in ThinkSpeak which is shown in figure 6.



Fig 6. Detecting social distancing

Furthermore, as shown in Figure 7, the number of people that violate social distancing is kept in the cloud as a graphical and numeric representation.

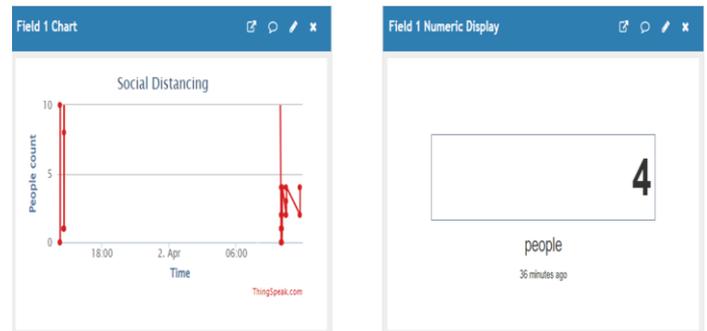


Fig 7. Count of people violating social distancing

In addition to this alert will also be sent to higher authorities / supervisor allotted to check for crowd and social distancing at a particular place. An audio alert will be given through the speaker automatically to the people that this number of people are violating social distancing.

A webcam is used to collect real-time footage, which is then fed into opencv-python. The accuracy level for real-time footage is relatively poor because it does not detect all the persons in the frame. To improve accuracy, videos that have been trained are used as input.



Fig 8. Person detection in real time video

The number of people detected through real time data capture inside a building is shown in Fig. 9.



Fig. 9. Thingspeak output of count of people

CONCLUSION

Distinguishing people helps with swarm control openly spaces. This is achieved with the assistance of PC vision and the Raspberry Pi. The Haar flood computation is used to check swarm size. The distance between every individual can be essentially assessed utilizing PC vision. Assuming that any arrangement of people is found abusing the base acknowledged limit esteem they will be demonstrated with a red bouncing box. A pre-shot video of individuals on a packed road is utilized in the created framework. Recordings are given as info and results are viewed as productive. The group size and social separating assessment can be by and large utilized for observing reason.

REFERENCES

- [1] Abbas, S. Syed Ameer, M. Anitha, and X. Vinitha Jaini. "Realization of multiple human head detection and direction movement using Raspberry Pi." In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1160-1164. IEEE, 2017. <https://doi.org/10.1109/WiSPNET.2017.8299946>
- [2] Rucha Visal, Atharva Theurkar, Bhairavi Shukla, "Monitoring Social Distancing for Covid-19 Using OpenCV and Deep Learning", *International Research Journal of Engineering and Technology (IRJET)* Volume: 07 Issue: 06, p-ISSN: 2395-0072, June 2020.
- [3] Ms. Subashree D, Shrushti Rohidas Mhaske, Sonal Rajesh Yeshwantrao, Ayush Kumar, "Real Time Crowd Counting using OpenCV", *International Journal of Engineering Research & Technology (IJERT)* ISSN: 2278-0181, Vol. 10 Issue 05, May 2021.
- [4] Badhan Hemangi, K. Nikhita, "People counting system using raspberry pi with opencv," *International Journal for Research in Engineering Application & Management (IJREAM)*, ISSN : 2494-9150 Vol-02, Issue 01, APR 2016. <https://doi.org/10.1109/ELINFOCOM.2016.7563020>
- [5] Kanchan Mangrule, H. T. Ingale, Vijay D. Chaudhari, Dr. A. J. Patil, "Literature Survey of Iot Capabled Crowd Analysis Using Raspberry Pi-3," *International Journal of Innovations in Engineering and Science*, Vol 4, No.10, 2019.
- [6] Md Israfil Ansari, Shim Jaechang, "People Counting System using Raspberry Pi", *Journal of Multimedia Information System* VOL. 4, NO. 4, pp. 239-242, December 2017.
- [7] A Jaysri Thangam, Padmini Thupalli Siva, B.Yogameena "Crowd Video Count In Low Resolution Surveillance Head Detector and Color based using Segmentation for Disaster Management", *IEEE ICCSP 2015 conference*.
- [8] Songyan Ma, Tiancang Du, "Improved Adaboost Face Detection," *International conference on measuring technology and mechatronics automation*, 2010. <https://doi.org/10.1109/ICMTMA.2010.184>
- [9] Min Li, Zhaoxiang Zhang, Kaiqi Huang and Tieniu Tan, "Estimating the Number of People in Crowded Scenes by MID Based Foreground Segmentation and Head-shoulder Detection," *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, IEEE*, 2008 <https://doi.org/10.1109/ICPR.2008.4761705>
- [10] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool, "Online Multiperson Tracking-by-Detection from a Single uncalibrated Camera", *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, September 2011. <https://doi.org/10.1109/TPAMI.2010.232>
- [11] Tao Zhao, Ram Nevatia, "Tracking Multiple Humans in Complex Situations", *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, September 2004. <https://doi.org/10.1109/TPAMI.2004.73>
- [12] G V Shalini, M Kavitha Margret, M J Sufiya Niraimathi, S Subashree, "Social Distancing Analyzer Using Computer Vision and Deep Learning", *Journal of Physics: Conference Series*, 2021. <https://doi.org/10.1088/1742-6596/1916/1/012039>