IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ANALYZING CC CAMERA FOR SOCIAL DISTANCE

Shabad Niveditha, Kasam Manideep Reddy, V. Bhanu Prakash, G. Sreenivasulu, Dr P. SrinivasaRao

Student, Student, Student, Associate Professor, Professor
Computer Science and Engineering
JB Institute of Engineering and Technology, Hyderabad, India

Abstract:

Social distancing (SD) is an effective measure to prevent the spread of the infectious Coronavirus Disease 2019 (COVID-19). However, a lack of spatial awareness may cause unintentional violations of this new measure. Against this backdrop, we propose an active surveillance system to slow the spread of COVID-19 by warning individuals in a region-of-interest. Our contribution is twofold. First, we introduce a vision-based real-time system that can detect SD violations and send non-intrusive audio-visual cues using state-of-the-art deep-learning models. Second, we define a novel critical social density value and show that the chance of SD violation occurrence can be held near zero if the pedestrian density is kept under this value.

***Index Terms* – Social distancing, YOLO v3, Anaconda Navigator.**

I. INTRODUCTION

The rapid outbreak of the Coronavirus Disease 2019 (COVID-19) has imposed restrictions on people's movement and daily life. Reducing the spread of the virus mandates constraining social interactions, traveling, and access to public areas and events. These limitations arise to mainly advocate social distancing; the practice of increasing physical space among people to minimize virus transmission. Monitoring and maintaining social distancing is carried out by governmental bodies and agencies using mass surveillance systems and closed-circuit television (CCTV) cameras. Nonetheless, this task is cumbersome and suffers from subjective interpretations and human error due to fatigue; hence, computer vision and machine learning tools are convenient for automation. In addition, they enable crowd behavior to be monitored and anomalies such as congested regions, curfew infractions, and illegal gatherings to be recognized. The widespread of mass surveillance and its integration with Machine Learning is hindered by ethical concerns, including possible breach of privacy and potential abuse. Therefore, privacy preserving surveillance and Machine Learning solutions are paramount to their ethical adoption and application.The design of vision based social distance estimation and crowd monitoring system deals with the following challenges

- Geometry understanding, in terms of ground plane identification and homo graph estimation;
- Multiple people detection and localization;
- Statistical/temporal characterization for social distance infractions, e.g., short term violations are irrelevant. Currently, Machine Learning-based solutions identify social distance infringements using off-the-shelf person detection and tracking models.

In general, the models' performance is conjoined with privacy; they yield high performance by carrying and processing person-specific information to develop robustness against occlusions and missing data. In addition, they localize human subjects via bounding boxesthat can be over-sized or incomplete which results in significant distance estimation errors.Therefore, we propose a privacy-preserving adaptive social distance estimation and crowd monitoring systemthat can be implemented ontopof anyexisting CCTV infrastructure.

Developing a robust person localization strategyusing pose estimation techniques; Forming an adaptive smoothing and tracking paradigm to mitigate the problem of occlusions and missing data without compromising privacy; Designing a real-time privacy-preserving social distance estimation and crowd monitoring solution with potential to cover other application areas and tasks.

II. LITERATURE SERVEY

Feasibility analysis begins once the goals are defined.It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing thingsgenerate new ideas. There is no need to go into the detailed system operation yet. The solutionshould provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exertconsiderable effort at this stage only to achieve the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are

Technical feasibility:

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it. Install all upgrades framework into the python package supported windows based application.

Operational Feasibility:

The system working is quite easy to use. Users require no special training for operating the system. Technical performance includes issues such as determining whether the system can provide the right information for the depressed individual with the least amount of false positives or false negatives, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services.

Economically Feasibility:

Development of this application is highly economically feasible. It is cost effective in the sense that it eliminated the paper work completely. The system is also time effective because the prediction is automated.

III. REQUIREMENTS

Functional Requirements:

Analyzing CC Camera For Social Distance involves the following functions

- Analyzing the Distance between the two peoples.
- Predict of distance.

Non-Functional Requirements:

Performance: Easy tracking of records and updating can be done. All the requirements relating to performance characteristics of the system are specified in the section below. There are two types of requirements.

Static Requirements:

These requirements do not impose any constraints on the execution characteristics of the System. They are: Number of Users: The number of users may vary, as this software finds applications in fit to use applications only.

Dynamic Requirements:

These specify constraints on the execution characteristics of the system. They typically include response time and throughout the system. Since these factors are not applicable to the proposed software, it will suffice if the response time is high and the transactions are carried out precisely and quickly.

Reliability:

The software will not be able to connect to the data file in the event that the path of the file isn't correct or in the event of the co lab server being down due to a hardware or software failure.

Availability:

The software will be available to anyone and everyone as the trained model can be integrated with any well working system and can produce the desired results.

Security:

There is very low necessity of security in this software as the trained model process is irreversible.

Maintainability:

Maintenance can be done by providing feedback on tweets whose prediction is incorrect.

Portability:

The software is a python trained model and can be deployed to any system with any operating system and decent RAM and memory.

IV. SYSTEM ARCHITEECTURE

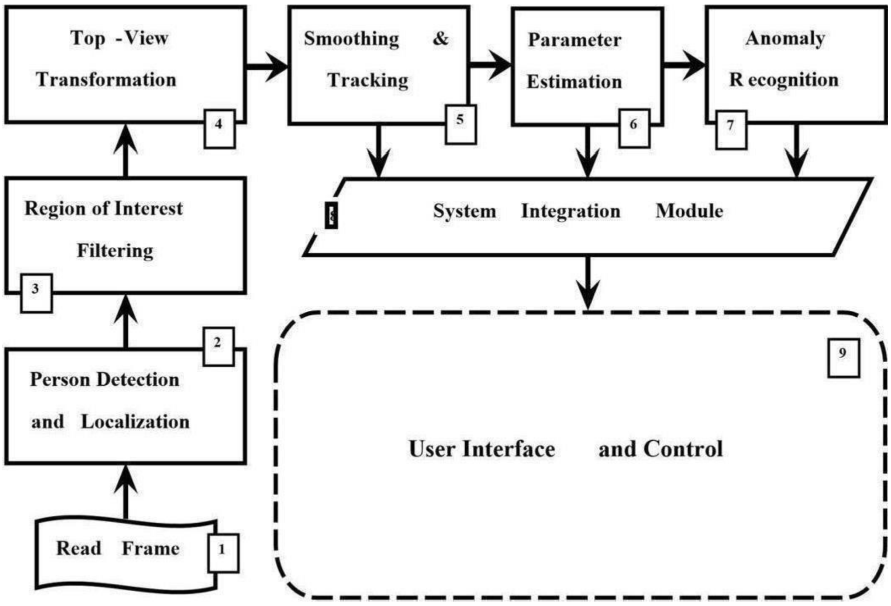

Fig : System architecture

V. ALORITHMS

Python

Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way or a functional way.

Yolo V3

YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images.YOLO uses features learned by a deep convolutional neural network to detect an object. YOLO is a Convolutional Neural Network (CNN) for performing object detection in real time. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage anaconda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions. The command-line program anaconda is both a package manager and an 13 environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

MODULES

Input Collection

The image captured and video recorded by the CCTV camera is given as the input. The camera is set up in a way it captures at a fixed angle and the video frame's view was changed into a 2D bird's view to accurately estimate the distance between each person. It is taken that the people within the frame are leveled on the horizontal plane. Then, four points from the horizontal plane are chosen, then it is changed into the bird's view. Now the position of each person can be calculated based on the bird's view.

Calibrating the Camera

The region of interest (ROI) of an image or a video frame focused on the person who is walking was captured using a CCTV camera was then changed into a two-dimensional bird's view. The changed view's dimension is 480 pixels on all sides. The calibration is done by transforming the view frame captured into a two-dimensional bird's view. The camera calibration is done straightforwardly using Open CV. The transformation of view is done using a calibration function that selects 4 points in the input image/video frame and then mapping each point to the edges of the rectangular two-dimensional image frame. On performing this transformation, every person in the image/frame is considered to be standing on a leveled horizontal plane.

Detection of pedestrians

Deep Convolutional Neural Networks model is a simple and efficient model for object detection. This model considers the region which contains only "Person" class and discards the regions that are not likely to contain any object. This process of extracting the regions that contain the objects only is called as Region Proposals. The regions predicted by region proposal can vary in size and can be overlapping with other regions. So to ignore the bounding boxes surrounding the overlapping region, depending upon the Intersection Over Union (IOU) score maximum non suppression is used.

Measurement Of Distance

The interval between the set of individuals in an input frame can be easily calculated once the bounding box for each person is mapped. To do so the bottom center of the box mapped to every person within the range is considered.

VI. DATASET

We utilize the EPFL-MPV, EPFL-Wild track, and OxTown public datasets along with the pose estimations prepared in The EPFL-MPV is comprised of four sequences, named 6pc0,6p-c1, 6p-c2, and 6p-c3, for six people moving freely in a room. The sequences are synchronized and view the same environment but from different perspectives. Each sequence is recorded at 25 frames per second (fps) and has 2954 frames. The EPFL-Wild track contains seven synchronized sequences, named C1-C7, with approximately 20 people moving outdoor. The sequences view walking pedestrians outside the main building of the ETH university in Switzerland. They are shot using seven cameras positioned at different locations and each has a total number of 400 frames. Lastly, the OxTown is a street surveillance video with 4501 frames shot with a single camera at 25 fps. It oversees, on average, 16 people walking down a street in Oxford, London. The utilized datasets offer annotations in terms of bounding boxes that localize people in the scene. Additionally, they provide the homography matrix and the image-to-real distance scale of each recording camera. The EPFL-MPV and OxTown bounding boxes are vertically over-sized and enclose more than the areas occupied by the human subjects. Therefore, their bottom mid-points are lower than the subjects actual ground positions. In this work, we correct for this by shifting the midpoints up a percentage of the bounding box total height. In specific, we apply a 10% and 2% uplift to the EPFLMPV and OxTown localization data, respectively. Moreover, the OxTown dataset annotation includes bounding boxes for babies in strollers/prams accompanied by adults. This is outside the 17 scope of our work; hence, we discard them (This corresponds to the following subject IDs:24, 42, 44, 45, and 47). Finally, the ROI for each dataset/sequence is manually selected, in the image–pixel domain, to cover the floor of the scene. The ROIs include most annotated positions, but we discard the remaining few that are outside the selected area. This corresponds to excluding 2.38% (960 out of 40,393), 6.67% (4767out of 71,460) and 15%(6403 out of 42,721) of the EPFLMPV, EPFL-Wild track, and OxTown annotations, respectively. The proposed system smoothing and tracking parameters are found for every dataset/sequence by minimizing the localization error in Equation (21) using the Bayesian optimization algorithm in MATLAB. The optimization is executed for 500 iterations using the expected improvement plus acquisition function and repeated five times for verification.

VII. METHODOLOGY

7.1 Object Detection Framework

Object Detection Framework To select the best-suited object detection model for our solution, we compared the performance metrics of several state-of-the-art models viz. YOLOv3, Faster R-CNN and SSD. Available benchmarks on COCO test-dev dataset demonstrate that the mean Average Precision (mAP) is comparable for the FRCNN and YOLOv3 models but much lower for SSD. On the other hand, both SSD and YOLOv3 process images at real-time speed (on a Pascal Titan X GPU) whereas FRCNN is considerably slower. Another important aspect of the people detection problem is to compare how different models perform under small and heavy occlusions. The benchmarking performed on the Euro City Persons test dataset using the log- average miss rate as a performance metric also demonstrates that YOLOv3 and FRCNN perform better than SSD. Considering the trade-off between all these metrics, finally YOLOv3 is chosen as the default object detector of our solution and is deployed without further training. 68.8 – 75.6 % and 75.5 – 85.1 % respectively. The best detection performance is observed with the pre-trained YOLOv3 detector for crowd sizes of up to 30 people, provided around 80% of each person was clearly visible.

7.2 Camera Calibration and Distance Estimation

The tool-based bird's eye view calibration process involves manual effort in the selection of the planar points for estimating the inverse perspective transformation, as well as the reference lengths. Such an approach, though relatively more accurate, is not convenient for deployment in enterprises with large numbers of cameras to be monitored. The automated calibration technique is easy to implement on a large scale; however, it occasionally exhibits errors in estimation, especially for people very close to the camera. Hence, it would benefit from restricting the detections to a predetermined region of interest or introducing some correction factors to improve the calculations. The automated calibration approach also expects the people detection module to be highly accurate as it relies on the bounding box information to compute camera parameters. In terms of distance estimation, both the techniques are quite fast, returning outputs in a fraction of a millisecond, thereby ensuring this module doesn't become a bottleneck in real-time processing. 21

7.3 Tracking and Compliance Metrics

As our application utilizes tracking-by-detection approach, the performance of the multi-object tracker (MOT) depends on the object detection model in use. MOT metrics obtained on the Oxford Town-center dataset with the YOLOv3 detector and motpy tracker like motpy. Moreover, the tracker has an update rate of 260 Hz which is ideal for real-time applications. Finally, we analyzed the variation in our proposed compliance metrics over time on the Town-center dataset. A bulk ofthe violations are found to be transient in nature, whereas a small proportion are persistent or high-risk violations. Also, we have observed that violations-to-violators ratio is higher for a clustered gathering compared to a queuelike formation and hence raises the risk oftransmission. 5.4 Results Model MAP FPS Occlusion.

VIII. TESTING

Model Testing

The proposed system operates in an automated way and helps to automatically perform the social distance inspection process. Once the model is trained with the custom data set and the pre-trained weights given, we check the accuracy of the model on the test dataset by showing the bounding box with the name of the tag and the confidence score at the top of the box. The proposed model first detects all persons in the range of cameras and shows a green bounding box around each person who is far from each other after that model conducts a test on the identification of social distances maintained in a public place, if persons breaching social distance norms bounding box color changes to red for those persons and simultaneously face mask detection is achieved by showing bounding boxes on the identified person's face with mask or non-mask labeled and also confidence scores. If the mask is not visible in the faces, and if the social distance is not preserved, the system generates a warning and send alert to monitoring authorities with face image. The system detects the social distancing and Mask with a precision score of 91.7% with confidence score 0.7, precision value 0.91 and the recall value 0.91 with FPS = 28.07.

System Integration

For integrating the proposed system outputs and displaying them on the user interface unit. These examples offer complementary interpretations for the scene and serve different purposes depending on the intended application or required analysis. For instance, the input video frame, depicted, is overlaid with the localization and averaged ODM results. This type of display is important when monitoring crowds in public areas or for analyzing customer's browsing habits and preferences in shops. Moreover, we show in Figure 6b that the former information can be replaced with the detected social distance violations and the averaged thresholded CDM. This example is directly intended for social distance monitoring applications and can be used to oversee critical waiting areas, e.g., in airports and hospitals. Furthermore, Figure 6c demonstrates a dynamic top-view map for the scene by plotting the localization, inter-personal distances, and the averaged CDM in the real-world coordinates.

Performance Evaluation

In order to test the performance of the proposed model, we used the Oxford Town Centre (OTC) dataset as a previously unseen and challenging dataset with very frequent cases of occlusions, overlaps, and crowded zones. The dataset also contained a good diversity of human specimens in terms of clothes and appearance in a real-world public place. In order to provide a similar condition for performance analysis of YOLO based models, we fine-tuned each model on human categories of the Google Open Images (GOI) data set. This was done by removing the last layer of each model and placing a new layer (with random values of the uniform probability distribution) corresponding to a binary classification (presence or absence of a human). Furthermore, in order to provide an equal condition for the speed and general is ability, we also tested each of the trained models against the OTC dataset. We evaluated and compared our developed models against three common metrics of object detection in computer vision, including Precision Rate, Recall Rate, and FPS against three state-of-the-art human/object detection methods. All of the benchmarking tests and comparisons were conducted on the same hardware and software: a Windows 10-based platform with an Intel© Core™ i5-3570K processor and an NVIDIA RTX 2080 GPU with CUDA version 10.1. In terms of mass deployment of the system, the above hardware setup can handle up to 10 input cameras for real-time monitoring of e.g., different floors and angles of large shopping malls. However, for smaller scales, a cheaper RTX 1080 GPU or an 8-core/16-thread 10th generation Core™ i7 CPU would suffice for real-time performance. illustrates the development of loss function in training and validations phases for four versions of our Deep SOCIAL model with different backbone structures. The graphs confirm a fast yet smooth and stable transition for minimising the loss function in DS version after 1090 epochs where we reached to an optimal trade-off point for both the training and validation loss. Table 3 provides the details of each backbone and the outcome of the experimental results against three more state-of-the-art model on the OTC Dataset. visualises the robustness of the proposed detectors in three challenging indoor/outdoor publicly available datasets: Oxford Town Centre, Mall Dataset, and Train Station Dataset. Interestingly, the Faster-RCNN model showed good general is ability; however, its low speed was an issue which seems to be due to the computational cost of the "region proposal" technique. Since the system required a real-time performance, any model with the speeds slower than 10 fps and/or a low level of accuracy may not be a suitable option for Social Distancing monitoring. Therefore, SSD and Faster-RCNN failed in this benchmarking assessment, despite their popularity in other applications. 36 In terms of both speed and accuracy.

Video Processing

We use OpenCV imagine the expectation brings about recordings. OpenCV upholds perusing surges of recordings from outside gadgets and documents from the nearby document framework. Given a prepared model on a veil discovery dataset, we anticipate that the output of the model should contain at any rate the accompanying fields: A variety of pictures utilized in the expectation and a variety of forecasts produced by the model, of tuples of the accompanying organization (a) x, y directions of the upper left corner of the jumping box, standardized to picture width and tallness. (b) x, y directions of the base right corner of the bouncing box, standardized to picture width and tallness. (c) a gliding point certainty levels (d) a number demonstrating the anticipated class A variety of name names the video source is perused as an inerrable stream of casings of pictures. Each casing of picture is passed into our model at their unique tallness and width (e.g., 1080 pixels wide, 1920 pixels high). Our model produces derivation results adjusting to the above design. We utilize the outcomes to draw the bouncing boxes, anticipating class names and certainty level for each recognized (face, face covers, face veils worn mistakenly) on this edge of picture. The drawn casing is then passed into a video encoder to be saved as a casing in the yield video. The outcome is another video with the above perceptions with MPEG-4 encoding. The info video isn't altered in any capacity Processing recordings with OpenCV adds overhead to display expectation. The overhead comes from perusing outlines from the info video, drawing the perceptions and composing the attracted casing to the yield video. Model is very performant, accomplishing 2 edges for every second on a humble double center Intel Xeon CPU at 1920×1080 goal.

IX. CONCLUSION

Real-time system to monitor the social distancing and using the proposed critical social density to avoid overcrowding. We are focused on giving imaginative, strategic advances that ensure individuals and networks. Implementing social separating measures while amidst a progressing worldwide pandemic is an upward fight that each district and business is confronting today. It has been sent to get ready associations to adjust to the new standard to encourage appropriate adherence to rules and keep each local area part protected and sound. This task has pragmatic worth under the current setting of the COVID 19 pandemic. Pipeline is now fit for recognizing individuals with, without and inaccurately wearing covers with sensible exactness. For certain enhancements, we imagine that item can be utilized as a segment in a contact following framework. Item is likewise generally Computationally effective. The equipment limit for sending is low. This implies that item is less confined by financial plan or the degree of monetary improvement at the area of its organization and henceforth can arrive at more places where COVID- 19 diseases present more danger to individuals.

X. FUTURE WORK

This method was developed with an efficient way for the people who are not wearing face mask and not maintaining social distance and notified to officials by email. As a future enhancement, we can predict/detect time at which it gets crowded and heat map can be plotted in an accurate was

ACKNOWLEDGMENT

This research was supported/partially supported by JB Insititue of Engineering and Technology. We are thankful to our technical and non technical members who helped me in doing the reserach, although they may not agree with all of the interpretations provided in this paper.

REFERENCES

[1] Fauci, A.S.; Lane, H.C.;Redfield, R.R. COVID-19—Navigating the Uncharted. N.Engl. J. Med. 2020, 382, 1268–1269

[2] Hoeben, E.M.; Bernasco, W.; Suonperä Liebst, L.; van Baak, C.; Rosenkrantz Social distancing compliance: A video observational analysis. PLoS ONE 2021, 16, e0248221.

[3] Hossain, M.S.; Muhammad, G.; Guizani, N. Explainable AI and Mass Surveillance System-Based Healthcare Framework to Combat COVID-I9 Like Pandemics. IEEE Netw.2020, 34, 126–132.

[4] Cristani, M.; Bue, A.D.; Murino, V.; Setti, F.; Vinciarelli, A. The Visual Social Distancing Problem. IEEE Access 2020, 8, 126876– 126886.

[5] Sugianto, N.; Tjondronegoro, D.; Stockdale, R.; Yuwono, E.I. Privacy-preservingAIenabled video surveillance for social distancing: Responsible design and deployment forpublic spaces. Inf. Technol. People 2021