Pandemic control measures and analysis using computer vision

Kavin Chandar Arthanari Eswaran [RA1811003010205],
Akshat Prakash Srivastava [RA1811003010157],
Gayathri M (Mentor)

Abstract

***Abstract*—** Objects are commonplace, and we intuitively understand their purpose and objectives. We know a coffee cup is a cup by how it looks, and we can differentiate an empty one from a filled one just from a glance. Championing this idea forward with new-age technologies, we create a system to detect and sort the status of objects. We custom tailor our model to quantify pandemic control measures such as face mask usage and social-distancing protocols using computer vision.

For face mask usage, we detect various facial landmarks and correlate them with face mask position to accurately determine the different states of wearing a mask in real-time and provide analytics on overall face mask usage.

For social-distancing protocols, we detect and count pedestrians in a controlled space from a forty-five-degree angled top-down camera, determine if they are in line with official norms in real-time, and provide statistics on the status of social distancing in a specific area to inspire reformative changes. We aim to contribute to control measures and analysis using computer vision regarding the pandemic.

Objectives

Face-mask recognition system has various use-cases in current society,

- From entry checks into convenience stores to super malls
- Monitoring face mask usage in controlled environments such as offices, schools, and university campuses
- On the go mobile systems

Hence our objectives,

- Determine and differentiate the different states of wearing a mask
- Accurately classify and analyze face-mask usage for an individual
- Ease of implementation on any device without compromising quality
- Improve detection confidence to recognize any face-masks and make sure only masks are recognized
- Pictorially representing the acquired data for clarity
- Send non-intrusive warning cues

Objectives

Social-distancing measure system

- Quantifies status of social distancing in controlled environments
- Helps make informed decisions to regulate crowds
- Count and organize crowds

Hence our objectives,

- Current systems are hardware intensive and require expensive equipment
- Accurately gauging the distance between people in a 3D plane
- Accurately detecting individuals in high density crowds
- Counting crowds and classifying them in real-time
- Compatibility with CCTV and other cameras

Methodology

Face-mask recognition system

- We use OpenCV to capture live video footage. We then go frame by frame through the video.
- Each frame is passed to a mask_prediction method() from which we gain the location and prediction results of a face mask if detected.
- We also pass frames to a face detector method from which we earn a list of all detected faces.
- Through this list of faces, we analyze each for facial landmarks such as nose, mouth, and eyes.
- We then correlate the prediction result of the mask detection model with the detected facial landmarks in real-time through a particular status method.
- This method allows us to properly distinguish and identify all possible states of a worn face mask.
- This method then returns the appropriate label and color to determine forms in real-time. Next, we allow the user to press ‘X’ at any time to pause the video and return the appropriate face-mask usage statistics up till that point.
- We pictorially represent this using matplotpylib to show the relational duration with a pie chart.

Finally, we also run tests to assess the program's overall performance to objectively address and improve upon challenges and limitations.

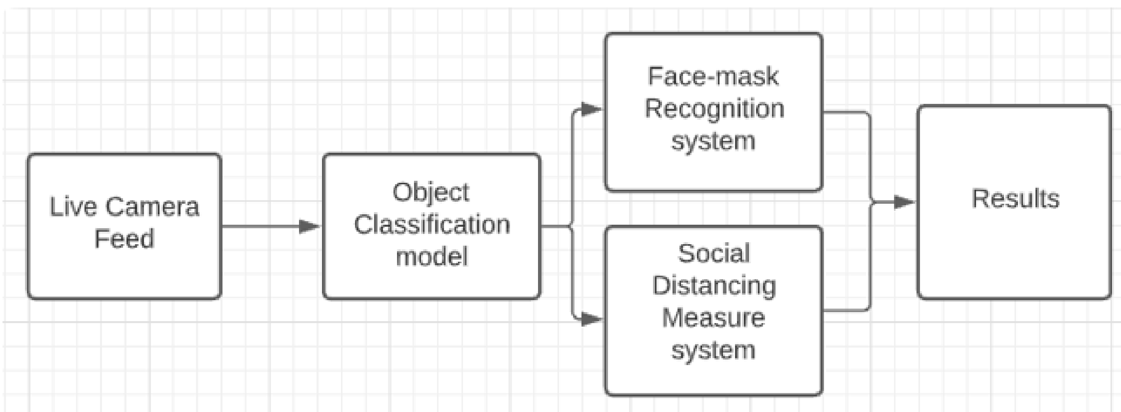
Methodology

Social-distancing measure system

- We take live video footage from CCTV cameras present at a forty-five-degree incline from the top.
- We then analyze this footage frame by frame in real-time to detect social-distancing violators.
- We first begin by detecting pedestrians by passing through a pedestrian_detection model.
- This method returns the confidence percentage, bounding box dimensions, and centroids of the detected pedestrians.
- We then give the list of centroids and their respective frames to the violation method.
- This method calculates the distance between the centroids to check for social-distancing violations and then returns a color to match the status.
- We also use this method to count crowds in a frame and classify them into followers and deviants in real-time.
- We pictorially represent this data using matplotlib—pyplot as a pie chart to indicate the social-distancing index.

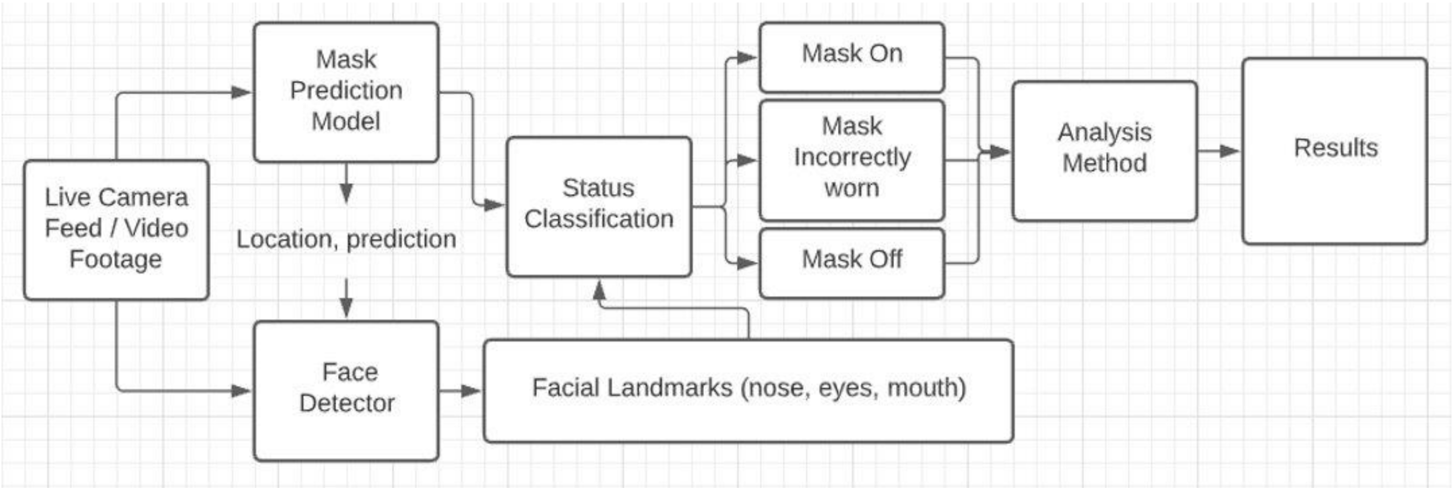
Finally, we run tests to test the system's limitations to optimize as best as possible.

Architecture Diagram



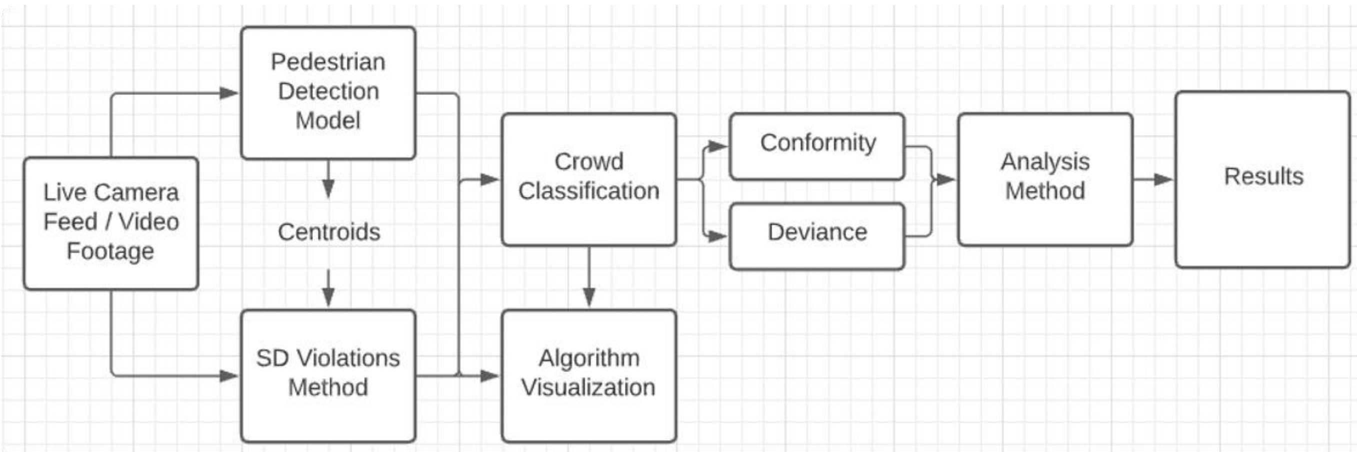
Base overview diagram

Architecture Diagram



Face-mask recognition system architecture diagram

Architecture Diagram



Social-distancing measure system architecture diagram

Modules Description

Face-mask recognition model

- **OpenCV** - we use openCV for live video capture and display purposes
- **Dlib** - primarily used to detect the 68 facial landmarks
- **Numpy** - convenient mathematical utility library
- **Matplotlib.pyplo**t - helps us plot graphs and pictorially represent statistics and analytics
- **Tensorflow – keras** - used to load model, preprocess input and change image to array
- **Mask_detector.model** - pre trained model helps us recognize a plethora of masks

Social-distancing measure

- **OpenCV** - we use openCV for live video capture and display purposes
- **Numpy** - convenient mathematical utility library
- **Imutils** – resize display window
- **Scipy.spatial** - used to calculate distance between centroids
- **Matplotlib.pyplot** - helps us plot graphs and pictorially represent statistics and analytics

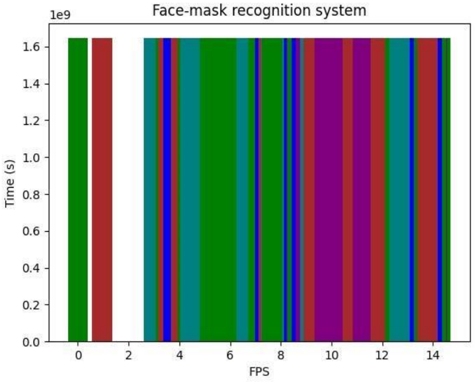
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Implementation

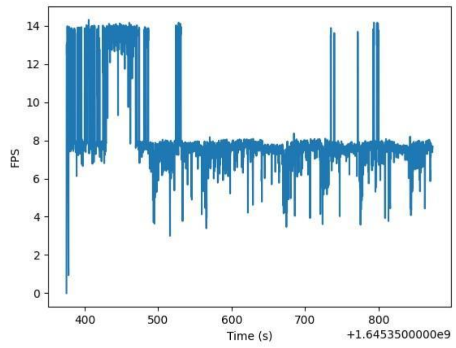
Results and Discussions

- Face-mask recognition

We can confidently distinguish the different states of a worn face-mask, Provide the statistical relationship between them, and pictorially represent them. However, the performance of our system averages at 8.8 fps and has immense fluctuations in the frame rate, which requires further optimization age.



Fps consistency

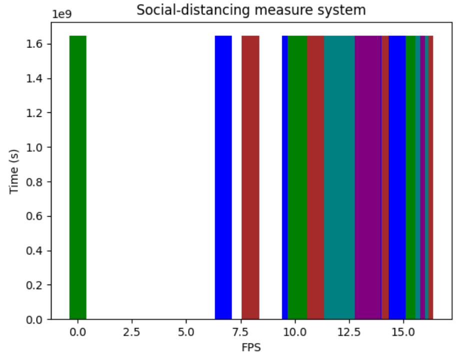


Fps drop rate

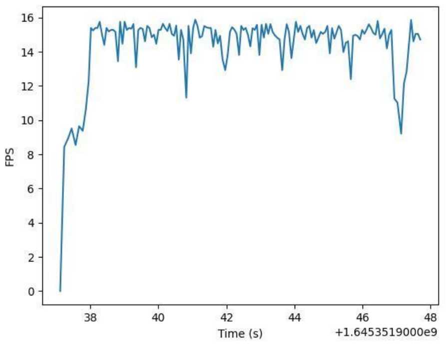
Results and Discussions

- Social-distancing measure

We can distinguish the different states of pedestrian movement into conformists and deviants accordingly. Provide the statistical relationship between them all and pictorially represent them. However, the performance of our system averages at nine fps and has immense fluctuations in the frame rate, which requires further optimization.



Fps consistency



Fps drop rate

Results and Discussions

S.No	Modules	Average Fps
1.	Object Classification model	16.0 Frames per second
2.	Face-mask Recognition system	8.8 Frames per second
3.	Social-distancing measure system	9.0 Frames per second

Synopsis of proposed models

Due to its software-only nature, It is incredibly cost-effective to layer onto other systems to check for face-mask usage and social-distancing awareness . Our particular system alone can be deployed onto every webcam to every high-rise CCTV camera system to monitor pandemic protocols . We genuinely believe it helps with making informed decisions in a post-pandemic society . From healthcare industries to commercial monopolies, Vision-based object status recognition can always be used to analyze and automate processes for various use-cases . There is immense potential for further improvement with Pandemic control measures and computer vision analysis.

Output

Output-face mask recognition system

Fig. 1.0 Mask On

Fig. 2.0 Mask Off

Fig. 3.0 Nose exposed

Fig. 3.1 Nose and Mouth exposed

Fig. 3.2 Eyes covered

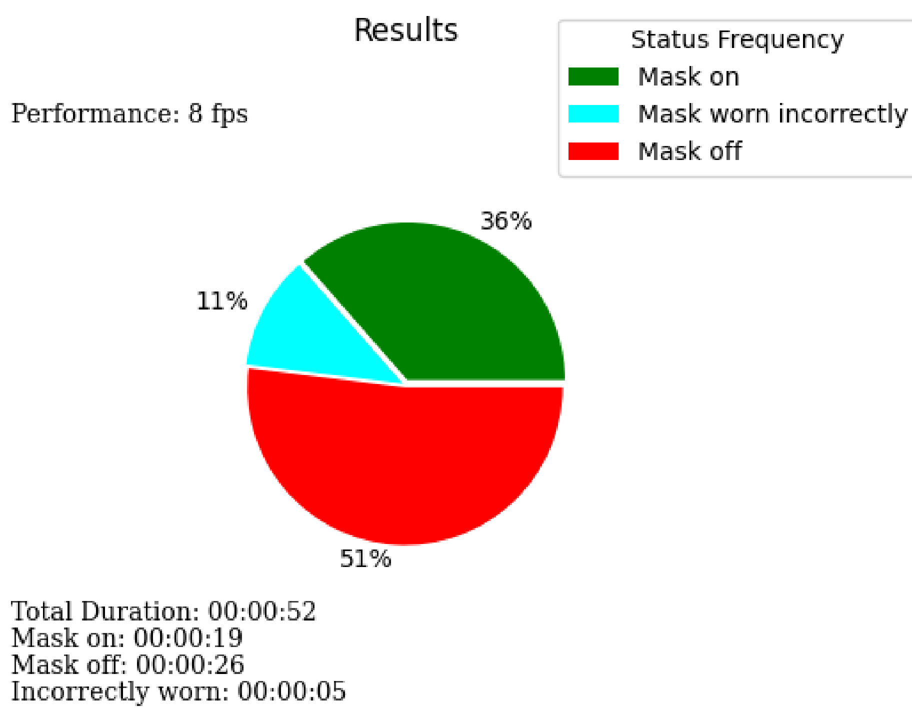


Fig 4.0 Face-mask Output Analytics

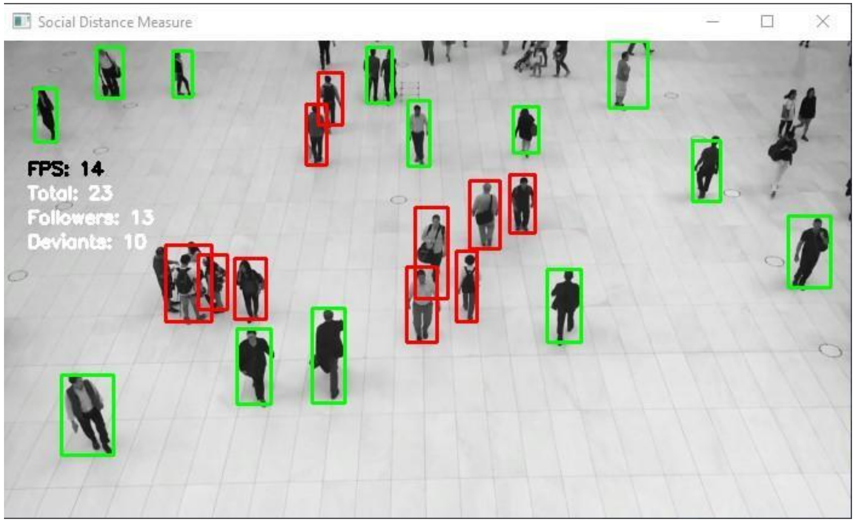


Fig. 5.0 Final output

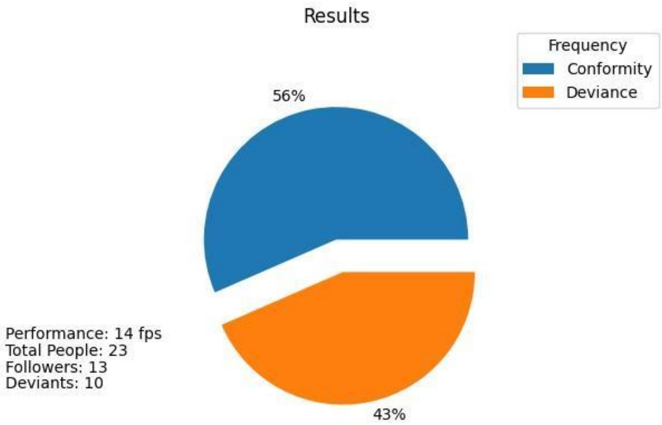


Fig. 5.1 Output Statistics

Fig. 6.0 Algorithm visualization

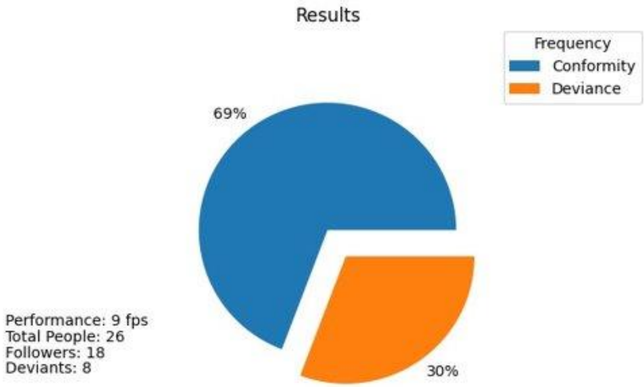


Fig. 6.1 Output Statistics

Literature Review

The vision-based face-mask recognition system

- [1-5] - These papers use methods such as the Deep Learning model (Inception V3) and achieves a 99.9 % rate of classification, Resnet 50 for feature extraction, Support vector classification, and ensemble algorithm for classification, SSDMNV2 for face detection, and MobilenetV2 for mask detection, One particular paper performs random sampling on MAFA dataset and achieves a p-value of 1.07 from 11.82, Retina Face Mask method uses new dataset for incorrectly worn masks.

Social-Distancing measure system

- [6-10]- These models use methods such as distance classification using YOLO's dataset paired with a thermal camera for temperature monitoring to identify SD Violators, Mobilenet, resnet, and VGG are used by another model to classify SD and face-mask violators.

Literature Review

Next, some models send auditory and visual warning cues to alert individuals and point out high-density areas. Finally, one impressive model uses Twitter's social mobility index, public geolocated Twitter data, to find SD violators. One particular paper finds the distance between pre-distributed mobile nodes in places like university campuses as a method.

Vision-based Object Sorting System,

- [11-15]-The following sorting systems specialize in particular fields, such as mango feature extraction using 2D and 3D visual properties. Next, we observe a vision-based book sorting algorithm that extracts book contours normalizes the image to sort books. We also see a unique prototypical computer vision-based date grading and sorting system. More widely used object sorting systems are robotic garbage sorting systems using RPN and VGG-16. Finally, we see a vision-based robot sorting system that uses STM32F4 as the central controller and CDD imaging sensor for the general sorting of targets using a robot.

References

1. Chowdary, G. Jignesh, et al. "Face mask detection using transfer learning of inceptionv3." *International Conference on Big Data Analytics*. Springer, Cham, 2020.
2. Loey, Mohamed, et al. "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic." *Measurement* 167 (2021): 108288.
3. Nagrath, Preeti, et al. "SSDMNV2: A real-time DNN-based face mask detection system using single shot multibox detector and MobileNetV2." *Sustainable cities and society* 66 (2021): 102692.
4. Sethi, Shilpa, Mamta Kathuria, and Trilok Kaushik. "Face mask detection using deep learning: An approach to reduce risk of coronavirus spread." *Journal of biomedical informatics* 120 (2021): 103848.
5. Jiang, Mingjie, Xinqi Fan, and Hong Yan. "Retinamask: A face mask detector." *arXiv preprint arXiv:2005.03950* (2020).
6. Saponara, Sergio, Abdussalam Elhanashi, and Alessio Gagliardi. "Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19." *Journal of Real-Time Image Processing* (2021): 1-11.
7. Teboulbi, Safa, et al. "Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention." *Scientific Programming* 2021 (2021).
8. Yang, Dongfang, et al. "A vision-based social distancing and critical density detection system for COVID-19." *Sensors* 21.13 (2021): 4608.
9. Xu, Paiheng, Mark Dredze, and David A. Broniatowski. "The Twitter Social Mobility Index: Measuring Social Distancing Practices With Geolocated Tweets." *Journal of medical Internet research* 22.12 (2020): e21499.
10. Kobayashi, Yutaro, et al. "A system for monitoring social distancing using microcomputer modules on university campuses." *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2020.
11. Chalidabhongse, Thanarat, Panitnat Yimyam, and Panmanas Sirisomboon. "2D/3D vision-based mango's feature extraction and sorting." *2006 9th International Conference on Control, Automation, Robotics, and Vision*. IEEE, 2006.
12. Gotoh, Toshiyuki, et al. "A flexible vision-based algorithm for a book sorting system." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.3 (1988): 393-399.
13. Al Ohali, Yousef. "Computer vision-based date fruit grading system: Design and implementation." *Journal of King Saud University-Computer and Information Sciences* 23.1 (2011): 29-36.
14. Zhihong, Chen, et al. "A vision-based robotic grasping system using deep learning for garbage sorting." *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017.
15. Zhang, Yawan, and Wei Cheng. "Vision-based robot sorting system." *IOP Conference Series: Materials Science and Engineering*. Vol. 592. No. 1. IOP Publishing, 2019

Plagiarism Report

Proof of Publication