VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Project Report on

Real Time Face Mask detection and Social distance Monitoring System

In partial fulfillment of the Third Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2020-21

Submitted by
Shreyas Kotkar D12C - 34
Yashkumar Jain D12B - 24
Chirag Kinger D12C - 31
Vikram Virwani D12B - 68

Project Mentor

Dr. (Mrs.) Sujata Khedkar

(2020-21)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
Department of Computer Engineering

Certificate

This is to certify that ***Shreyas Kotkar, Yashkumar Jain, Chirag Kinger & Vikram Virwani*** of Third Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the mini project on "***Real Time Face Mask detection and Social distance Monitoring System***" as a part of their coursework of Mini Project for Semester-VI under the guidance of their mentor ***Prof. Dr. (Mrs.) Sujata Khedkar*** in the year 2020-21.

This mini project report entitled (***Title***) by (***Author Name***) is approved for the degree of _____________ (***Degree details***).

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide: Internal and External

Mini Project Report Approval
For
T. E (Computer Engineering)

This mini project report entitled "***Real Time Face Mask detection and Social distance Monitoring System***" by ***Shreyas Kotkar, Yashkumar Jain, Chirag Kinger & Vikram Virwani*** is approved for the degree of T.E Computer Engg.

Internal Examiner

External Examiner

Head of the Department

Principal

Date:
Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

--	--
Shreyas Kotkar	Yashkumar Sudesh Jain
--	--
D12C-34	D12B-24
--	--
Vikram Virwani	Chirag Kinger
--	--
D12B-68	D12C-31

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor Dr. (**Mrs.) Sujata Khedkar** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement at several times.

Computer Engineering Department

COURSE OUTCOMES FOR T.E Mini Project

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one’s competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Abstract

The COVID-19 pandemic forced governments across the world to impose lockdowns to prevent virus transmissions. This resulted in the shutdown of all economic activity and accordingly the production at manufacturing plants across most sectors was halted. While there is an urgency to resume production, there is an even greater need to ensure the safety of the workforce at the plant site. Reports indicate that maintaining social distancing and wearing face masks while at work clearly reduces the risk of transmission.We decided to use computer vision on CCTV feeds to monitor worker activity and detect violations which trigger real time alerts on the shop floor. This paper describes an efficient and economic approach of using AI to create a safe environment in a manufacturing setup. We demonstrate our approach to build a robust social distancing measurement algorithm using a mix of modern-day deep learning and classic projective geometry techniques. We have also described our face mask detection approach which provides a high accuracy across a range of customized masks.

The end of 2019 witnessed the outbreak of Coronavirus Disease 2019 (COVID-19), which has continued to be the cause of plight for millions of lives and businesses even in 2020.As the world recovers from the pandemic and plans to return to a state of normalcy, there is a wave of anxiety among all individuals,especially those who intend to resume in-person activity. Studies have proved that wearing a face mask significantly reduces the risk of viral transmission as well as provides a sense of protection. However, it is not feasible to manually track the implementation of this policy. Technology holds the key here. We introduce a Deep Learning based system that can detect instances where face masks are not used properly. Our system consists of a dual-stage Convolutional Neural Network (CNN)architecture capable of detecting masked and unmasked faces and can be integrated with pre-installed CCTV cameras. This will help track safety violations, promote the use of facemasks, and ensure a safe working environment.

The CNN algorithm and Dataset allotment for mask detection. This paper manages complex pictures using facial recognition packages. The CNN methodology used in the security system and the medical system. The proposed work balanced face restriction, color changes, brightness changes, and contrast changes. Segmentation and feature extraction used in face restriction of the person image. We chose the CNN algorithm for Mask detection and Social distance. Regions with Convolutional neural network Based on Mixing pictures, pixel prediction, and specific enhancements. The main objective was to solve multiple and multitask picture detection problems with speed rates. The Methodology used for face detection and detection of Unmask person in a dataset of face database.

Keywords :- machine Learning Algorithm,Tensorflow , OpenCV, numpy, sklearn, Argparser, Imutils and YOLOv3.

Index

Title **Page no**

Chapter 1 - Introduction

In this chapter we will be talking about the problem statement in detail as well as the motivation behind our project and finally a brief introduction to the methodology we are going to use.

1.1 - Introduction to the project

The spread of COVID-19 Pandemic Disease has created a most crucial global health crisis of the world that has had a deep impact on humanity and the way we perceive our world and our everyday lives. In December 2019 the spread of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), a new severe infectious respiratory disease emerged in Wuhan, China and has infected 7,711 people and 170 reported deaths in China before coronavirus was declared as a global pandemic, was named by the World Health Organization as COVID-19 (coronavirus disease 2019). According to the World Health Organization (WHO as of April 16,2021) report the current outbreak of COVID-19, has infected over 140,121,962 people and more than 3,004,963 deaths in more than 200 countries around the world, carrying a mortality of approximately 3·4%, compared with a mortality rate of less than 1% from influenza.

A novel coronavirus has resulted in person-to-person transmission but as far as we know, the transmission of the novel coronavirus causing coronavirus disease 2019 (COVID-19) can also be from an asymptomatic carrier with no covid symptoms. Till now there is no report about any clinically approved antiviral medicine or vaccines that are effective against COVID-19. It has spread rapidly across the world, bringing massive health, economic, environmental and social challenges to the entire human population. At the moment, WHO recommends that people should wear face masks to avoid the risk of virus transmission and also recommends that a social distance of at least 2m be maintained between individuals to prevent person-to person spread of disease. Furthermore, many public service providers require customers to use the service only if they wear masks and follow safe social distancing. Therefore, face mask detection and safe social distance monitoring has become a crucial computer vision task to help the global society.

This paper describes an approach to prevent the spread of the virus by monitoring in real time if a person is following safe social distancing and wearing face masks in public places. This paper adopts a combination of YOLOv3 and tensorflow with transfer learning technique to achieve the balance of resource limitations and recognition accuracy so that it can be used on real-time video surveillance to monitor public places to detect if persons wearing face masks and maintaining safe social distancing. Our solution usesYOLOv3 models to analyze Real-Time Streaming Protocol (RTSP) video streams using OpenCV. We mix the approach of modern-day deep learning and classic projective geometry techniques which not only helps to meet the real-time requirements, but also keeps high prediction accuracy. If the person is detected as not following the covid-19 safety guidelines, a red frame will be shown so the person using our application can maintain social distance from the crowd. It allows automating the solution and enforces the wearing of the mask and follows the guidelines of social distancing. This model was created to run on raspberry pi4 and the accuracy obtained was between 85%.

1.2 - Motivation

In the context of transmitted virus between humans by sputtering (spraying), wearing the mask on the face appears necessary to protect people and to limit the propagation of the disease. Currently, we are facing the 2019–2020 coronavirus pandemic. Coronavirus disease 2019 (COVID-19) is an infectious disease with first symptoms similar to the flu. The symptom of COVID-19 was reported first in China and very quickly spreads to the rest of the world. The COVID-19 contagiousness is known to be high by comparison with the flu.

Although many people are already convinced of the interest for wearing face protection mask such as suggested by the World Health Organization and scientific studies , one can observe that many individuals do not correctly wear their masks (see various mask wearing configurations). These observations have led nurses and other citizens to initiate prevention campaigns related to the public health education in wearing the mask.

Besides, the detection of mask through camera acquisition systems has also been investigated. In this context, applications are developed for detecting the presence of mask or not for the counting of individual wearing mask towards carrying out crowd statistics and even for facial identification of people wearing mask . Most of research systems in favor of the fight against COVID-19 are focused on people monitoring.

1.3 - Problem Definition

In the current world, the COVID-19 has taken many lives and till date it is spreading rapidly. Not only old people but young ones are also getting infected by the virus and losing their lives. There are no such systems in the market that can help us detect the social distancing or whether a person is wearing a mask or not. It is the main reason for the spread of COVID-19. For such a situation a system or an application needs to be created specially for public places like schools, colleges or shopping malls so that using this application will help the administrator in maintaining the social distancing effectively and the chain of spread of virus can be broken.

1.4 - Relevance of the Project

The basic aim of the project is to detect the presence of a face mask on human faces and whether they are maintaining social distancing or not on live streaming videos . Alongside this, we have used basic concepts of transfer learning in neural networks to finally output presence or absence of a face mask in an image or a video stream.

1.5 - Methodology used

The proposed system uses a transfer learning approach to performance optimization with a deep learning algorithm and a computer vision to automatically monitor people in public places with a camera integrated with a raspberry pi4 and to detect people with mask or no mask.

We have used the MobileNetV2 architecture as the core model for detection, as MobileNetV2 provides a huge cost advantage compared to the normal 2D CNN model.

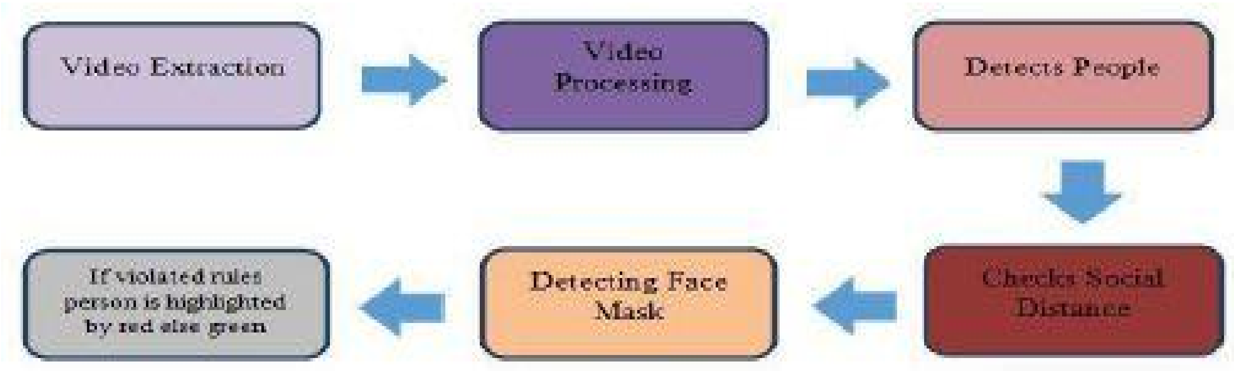


Fig 1 : Flowchart of Methodology used

Chapter 2 - Literature Survey

In this chapter we are going to tell you about the research we have done to implement the software. What we learned from them also we searched for some of the patents related to the topic.

A. Brief Overview of Literature Survey

Sr. No.	Title	Algorithm Used	Year
1	Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning *by* Adrian Rosebrock	OpenCV and Keras/TensorFlow and Deep Learning	05-2020
2	A Vision-based Social Distancing and Critical Density Detection System for COVID-19	AI, machine learning and computer vision	07 - 2020
3	Algorithm for Face Detection Combining Geometry Constraints and Face-Mask	combines geometry constraints and face-mask	06 - 2013
4	Face Recognition with Facial Mask Application and Neural Networks	DCT, MLP and RBF Neural Network	06-2013
5	Enabling and Emerging Technologies for Social Distancing: A Comprehensive Survey	AI, machine learning, computer vision, ultrasonic etc.	05-2020
6	Face Mask Detection using Transfer Learning of InceptionV3	RCNN, SSD, YOLO	07-2020

Sr No.	Title	Algorithm Used	Year
7	A Review on Face Mask Detection using Convolutional Neural Network	MobileNet V2 classifier, SVM, PCA	11-2020
8	Applying deep learning algorithm to maintain social distance in public place through drone technology	YOLO-v3, Darknet-53	05-2020
9	Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence	RCNN, MobileNet V2 classifier, SSD, YOLO-v3, Computer Vision	12-2020
10	Real-Time Face Mask Identification Using Facemask Net Deep Learning Network	Facemask Net, ReLu	07-2020

B. Related Works

2.1 - Research Papers Referred

a] Abstract of the research paper

While implementing our project we referred to some of the IEEE searched papers. the heading of the paper is mentioned in the above Literature survey table respectively.

1] Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning by Adrian Rosebrock :-This paper basically helped us in making face masks with OpenCV and Keras/TensorFlow. The author also discussed a two-phase COVID-19 face mask detector, detailing how our computer vision/deep learning pipeline will be implemented. According to his model the accuracy is around 95-99%.

2] A Vision-based Social Distancing and Critical Density Detection System for COVID-19 :- this paper tells us to use AI technology to calculate the 6 feet(2 meter) distance. Basically it works on 4 ethics :- (1) the system should never record/cache data, (2) the warnings should not target the individuals, (3) no human supervisor should be in the detection/warning loop, and (4) the code should be open-source and accessible to the public.

3] Algorithm for Face Detection Combining Geometry Constraints and Face-Mask :-this article mainly focuses on how the face mask should be applied.Basically for a small face some models were unable to detect. So this method combines geometry constraints and face-mask. The candidate faces can be detected by the geometry constraints between face and hair.

4] Face Recognition with Facial Mask Application and Neural Networks :-In this paper, it proposes a very fast image pre-processing by the introduction of a linearly shaded elliptical mask centered over the faces. Used in association with DCT, for features extraction, and MPL and RBF Neural Networks, for classification.

6] Face Mask Detection using Transfer Learning of InceptionV3 :-The next paper was Face Mask Detection using Transfer Learning of InceptionV3 in this author provided a solution on maintaining social distance. Faster RCNN, SSD, and YOLO v3, where YOLO v3 illustrated the efficient performance with balanced FPS and mAP score.and accuracy score of 87-93%.

7] A Review on Face Mask Detection using Convolutional Neural Network :- This was 7th paper used for reference by our grp in which author gave a solution using mobilenet v2 classifier and by using SVM analyzed image is fake or not and PCA used for feature extraction.

8]Applying deep learning algorithm to maintain social distance in public place through drone technology :- Applying deep learning algorithm to maintain social distance in public place through drone technology in this author gave a solution for maintaining social distance in public places with the help of drone.They trained a model using YOLO v3. And got an accuracy score of around 90-95%

9]Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence :- In this Mr shashi Yadav author of paper had given a solution for both face mask and social distance monitoring system. They have used the MobileNetV2 architecture as the core model for detection. Also with the help of SSD bound a box to a person if caught in video. They got an accuracy score of around 88-92%.

b] Inference drawn

Our Model is a combination of both face mask detection and social distancing monitoring , unlike other models that we have seen earlier which are either face mask detector or monitor social distancing. And also the accuracy of our model in detecting face masks is more than 98% while other models show an accuracy of only 80-90%.

2.2 - Patent search

1] Applicant Micronet Co., Ltd. Inventor ITO, Kazuhiko :-

A face recognition technology that enables high-speed determination of erroneous detection of face regions and high-precision face recognition by means of a first face recognition processing unit of relatively low accuracy for detecting a face region of a person on each captured image captured by a monitor camera, and a second face recognition processing unit for performing face recognition of a person more accurately with respect to the detected face region transmitted from the first face recognition processing unit. The second face recognition processing unit determines that the first face recognition processing unit has erroneously detected when the second face recognition processing unit is not able to recognize a face region on the detected face region transmitted from the first face recognition processing unit, and the first face recognition processing unit calculates and self-learns a barycentric coordinate of the detected face region as a point mask position. Thereafter, the first face recognition processing unit checks whether or not the coordinate of the barycentric position has already been stored in a storage as one of the point mask position, and, when the coincident coordinate of the point mask position is found in the data of the barycentric coordinates, this face recognition system does not perform more accurate face recognition processing by the second face recognition processing unit, and proceeds to the next face recognition processing on the next captured image

2] Applicant HANWANG TECHNOLOGY CO., LTD. Inventor HUANG LEI:-

The invention provides a face recognition method and belongs to the biological feature recognition field. With the method adopted, the problem of low efficiency of face recognition in the prior art can be solved. The method includes the following steps that: after it is detected that a camera is in a turned-on state, the brightness of a display screen is adjusted to a maximum value; a face image before the display screen is acquired; whether the face image meets a face recognition requirement is judged, if the face image does not meet the face recognition requirement, nonlinear transformation processing is performed on the face image, so that a face image meeting the face recognition requirement can be obtained; and the face image meeting the face recognition requirement is recognized. According to the face recognition method of the invention, when an electronic device is in a photographing mode, the brightness of the display screen is automatically adjusted to the maximum value, so that adequate illumination can be provide for face photographing, and therefore, face recognition efficiency can be improved; and the nonlinear transformation processing is performed on the face image, and therefore, the quality of the obtained face image can be improved, and face recognition accuracy can be improved.

3] Applicant Lovely Professional University Inventor Lovi Raj Gupta :-

A system to monitor the quarantine patients in the isolation wards for any unusual activity such as disobeying social distancing, trying to fleet away, restlessness, etc and tracks the patients activity in real-time. The real time monitoring data is collected and store in the cloud server for further analysis. Individual camera with unique identification number generated for each person records the activity and generates an alert if the person tries to flee or disobey social distancing norms.

2.3 - Inference drawn

- ❖ In all the patents that we have seen, there is no such project which detects face mask and social distancing as well .

- ❖ We took some inspiration from the Social distancing project made by Lovi Raj Gupta during making of our project.

2.4 - Comparison with the existing system

❖There is no such system available in India to be used in school or college campuses or public places.

❖All the existing systems are either face mask detector or social distancing manager but our model is a combination of both.

Chapter 3 - Requirement Gathering for the Proposed System

In this chapter we are going to discuss the resources we have used and how we analysed what the customer actually needs and what we can provide. We will also discuss the functional and non-functional requirements and finally the software and hardware used.

3.1 - Introduction to Requirement Gathering

Requirements gathering is an exploratory process that involves researching and documenting the project's exact requirements from start to finish. Effective requirements gathering and requirements management start at the beginning of the project. In our project we have targeted specially schools and colleges and the rest is given below.

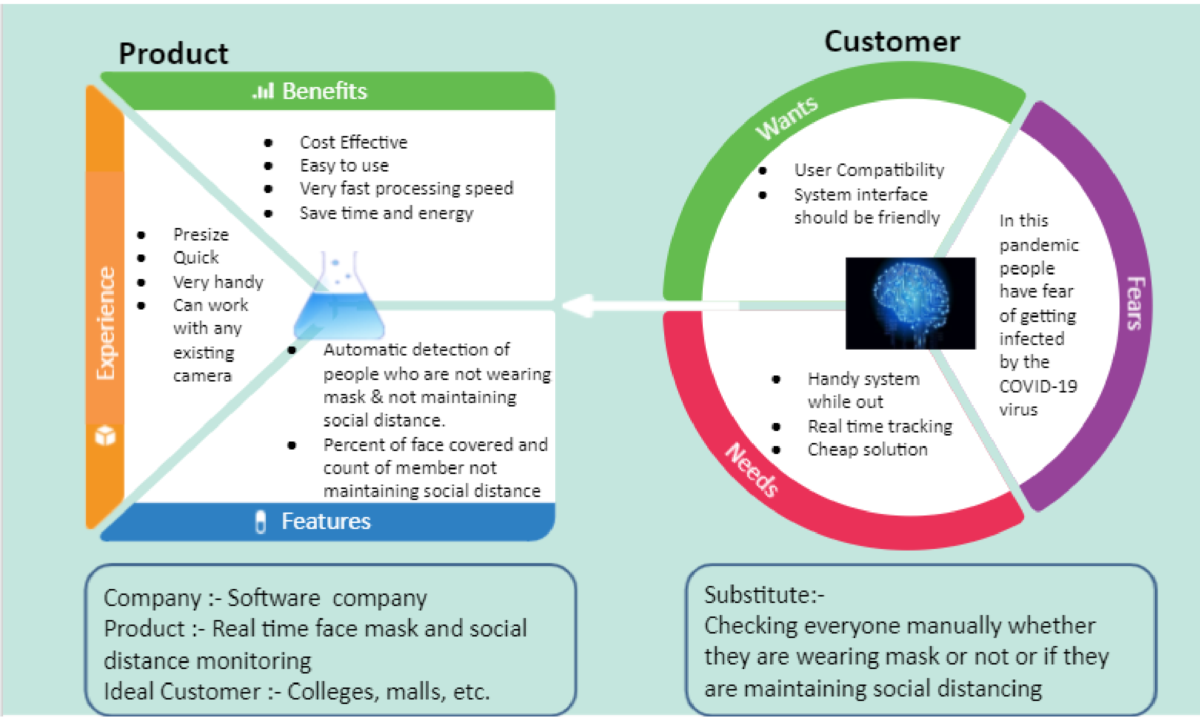


Fig 2 : Requirement Gathering in a VPC template

3.2 - Functional Requirements

Face Mask Detection uses existing IP cameras and CCTV cameras combined with Computer Vision to detect people without masks. Face Mask Detection Platform uses Artificial Intelligence Network to recognize if a user is not wearing a mask. The algorithm will run on 6SS IVA® Server which will be analyzing the streams coming from Milestone recording server and sending back MetaData which will allow the Smart Client user to receive No-mask data and alerts with the picture and the video of the person.

The Smart Client user can also benefit from 6SS Get Face® which is a module inside 6SS IVA® in order to identify the person who is not wearing a mask and send him an alert over his smartphone. If the camera captures an unrecognized face, a notification can be sent out to the administrator in order to take the appropriate action.

3.3 - Non-Functional Requirements

This project is intended to meet the following non functional requirements: -

1) This face recognition software should be available on the Internet, to enable the users to use , download it any time.

2) The program should be platform independent.

3) In this pandemic situation the application should be open for all.

3.4 - Hardware & Software Requirements

Software

Jupyter Notebook :- JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

Python 3.x (3.8 or earlier) :- python 3.x is the latest python software used in the companies where it is easy to use and helps us import the different build in libraries.

Fig 3 : Software Used

Hardware

GPU(Graphics Processor (NVIDIA) min 2GB) :- GeForce is a brand of graphics processing units (GPUs) designed by Nvidia. As of the GeForce 30 series, there have been seventeen iterations of the design. Most recently, GeForce technology has been introduced into Nvidia's line of embedded application processors, designed for electronic handhelds and mobile handsets.

Camera(CCTV/ Webcam/ Mobile Camera) :- The most important hardware of our model is the camera. Customers can use any type of camera he/she wants so that they can ensure their safety.

Storage Disk SSD(Min 400MB/s Read Speed) :- SSD adoption began in high-performance technology areas and in enthusiasts' PCs, where the drives' extremely low access times and high throughput justified the higher cost. But they have since become an accepted option -- or even the default choice -- in lower-cost mainstream laptops and PCs.

Fig 4 : Hardware Used

3.5 - Constraints

Many governments do not have a law that governs data.Critics of mask recognition also think that this new technology could be prone to some of the same pitfalls as facial recognition. Many of the training datasets used for facial recognition are dominated by light-skinned individuals. In 2019 Joy Buolamwini, a researcher at the Massachusetts Institute of Technology's Media Lab, and the AI Now Institute's Deborah Raji investigated the accuracy of commercially available datasets used by major tech companies . When they checked the performance of recognition systems using an algorithm trained with the standard datasets, and then using a new set of faces with much more racial and ethnic balance, the researchers found that the algorithm was less than 70 percent accurate in identifying new faces.

Thieves appear to have capitalized on this trend. They became eager early adopters, using surgical masks during the commission of various crimes, seemingly recognizing an opportunity weeks before the Centers for Disease Control on April 3 issued its first official recommendation to cover your face when going outside. Face masks are so coveted at this point they can sell on the black market at premiums of up to 1,500%.

Chapter 4 - Proposed Design

It is one of the main topics in this we have explained the methodology using the modular diagram and block diagram. Also we have shown the working timeline of our project using the Gantt chart.

4.1 - System Design / Conceptual Design (Architectural)

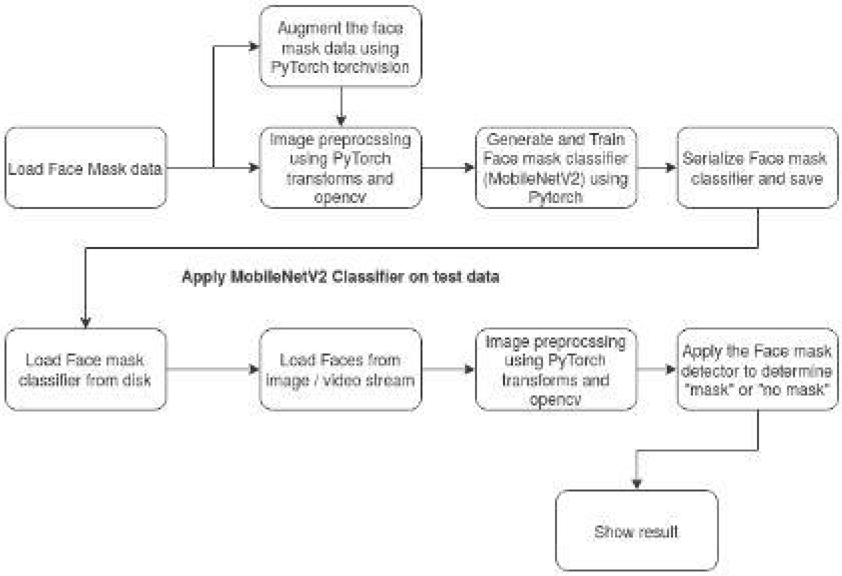


Fig 5 : System Design of our topic

4.2 - Modular Diagram

Here we initialize the video to our model which will simply extract the images from the video frame and apply a face mask classifier to each face. After this image processing it converts results into frames and displays face with box identification as shown in the modular diagram.

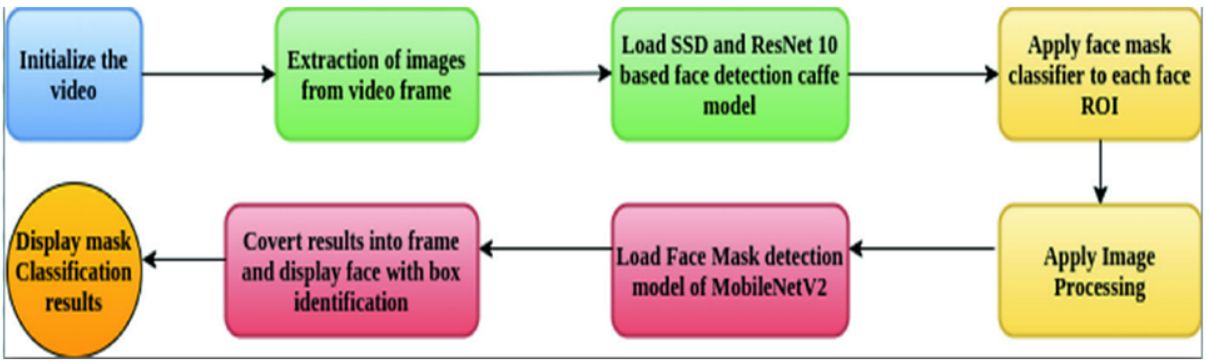


Fig 6 : Modular Diagram

4.3 - Detailed Design (DFD, Flowchart, State Transition Diagram, ER Diagram, etc...)

Face Mask detector

Training: Here we'll focus on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk.

Deployment: Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with mask or without mask as shown in the block diagram.

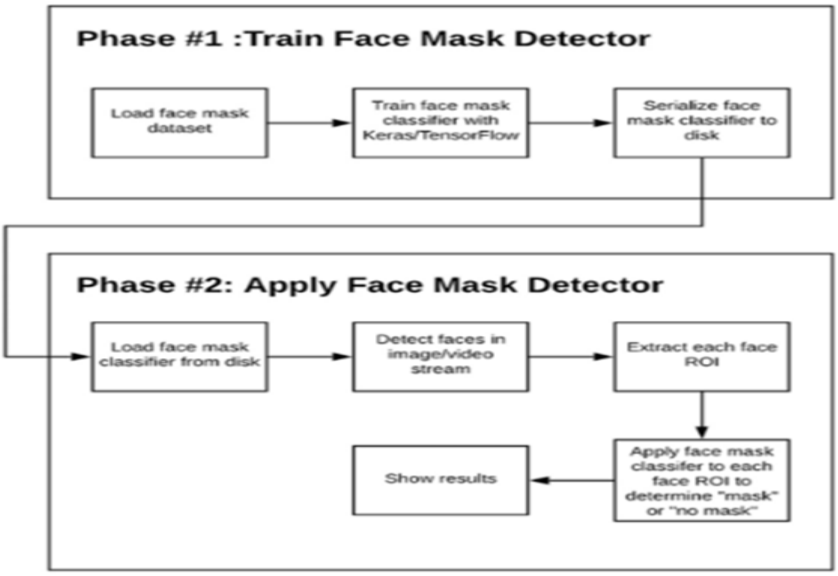


Fig 7 : Block diagram for Face mask detection

Social distance detector

Here, we are giving input from cctv or webcam which will pass to our pretrained algorithm which includes a custom social distance dataset. This pretrained algorithm predicts the images and labels them. In this way it displays the output with mask and social distance labeled on images as shewn below.

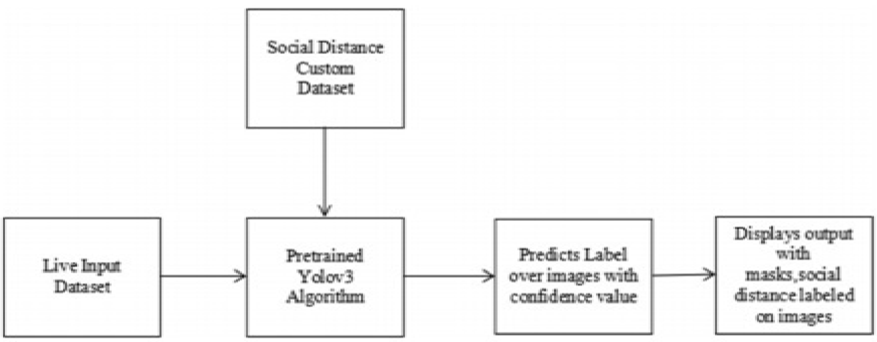


Fig 8 : Block diagram for Social Distance Monitoring

4.4 - Project Scheduling & Tracking using Timeline / Gantt Chart

Gantt chart of our project where how we worked for the whole semester to create this model is shown in a timeline pattern. It is the most important part to think and design the planning of your topic and so we planned our work like the gantt chart shown.

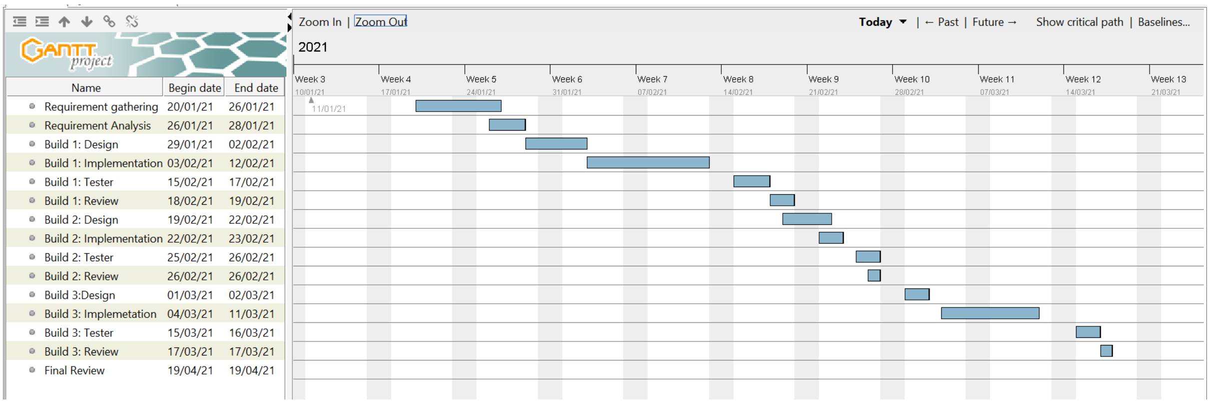


Fig 9 : Gantt chart

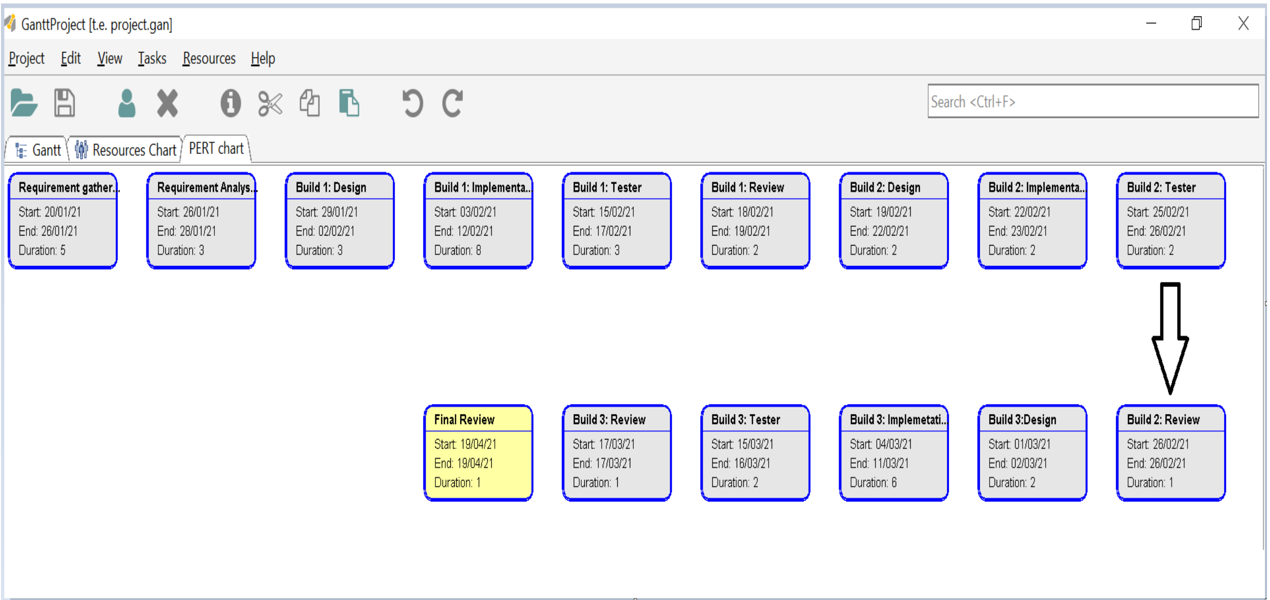


Fig 10 : PERT Chart

Chapter 5 - Implementation of the Proposed System

The most important chapter is the methodology we used. We have explained in detail how we build our model and we have shown the screenshots of our code as well as the output.

5.1 - Methodology applied

The system uses a transfer learning approach to performance optimization with a deep learning algorithm and a computer vision to automatically monitor people in public places with a camera integrated with a raspberry pi4 and to detect people with mask or no mask.

We have used the MobileNetV2 architecture as the core model for detection, as MobileNetV2 provides a huge cost advantage compared to the normal 2D CNN model. We are loading the MobileNet V2 with pre-trained ImageNet weights, leaving the network head off and constructing a new FC head, attaching it to the base instead of the old head, and freezing the base layers of the network.

Real-time person detection is done with the help of Single Shot object Detection (SSD).A bounding box will be displayed around every person detected. We after that calculate distance between all the persons detected in video. If the distance between two people is less than 2 meters, a red bounding box is shown around them, indicating that they do not maintain a social distance. We used custom face crop datasets of about 1400 images annotated in mask and no mask created by prajna bhandari.

The result of the SSD model extracts a person mask and displays a bounding box. The system monitors public places continuously and when a person without a mask is detected his or her face is captured and an alert is sent to the authorities with face image and at the same time the distance between individuals is measured in real time. Deploying our model to edge devices for automatic monitoring of public places could reduce the burden of physical monitoring, which is why we choose to use this architecture.

For calculating the social distance the most important part was to calculate the distance between to human beings. For calculating the distance we used the YOLOv3 module. The YOLOv3 algorithm generates bounding boxes as the predicted detection outputs. Every predicted box is associated with a confidence score. In the first stage, all the boxes below the confidence threshold parameter are ignored for further processing.

The rest of the boxes undergo non-maximum suppression which removes redundant overlapping bounding boxes. Non-maximum suppression is controlled by a parameter nmsThreshold. You can try to change these values and see how the number of output predicted boxes changes.

Next, the default values for the input width (in Width) and height (inpHeight) for the network's input image are set. We set each of them to 416, so that we can compare our runs to the Darknet's C code given by YOLOv3's authors. You can also change both of them to 320 to get faster results or to 608 to get more accurate results.

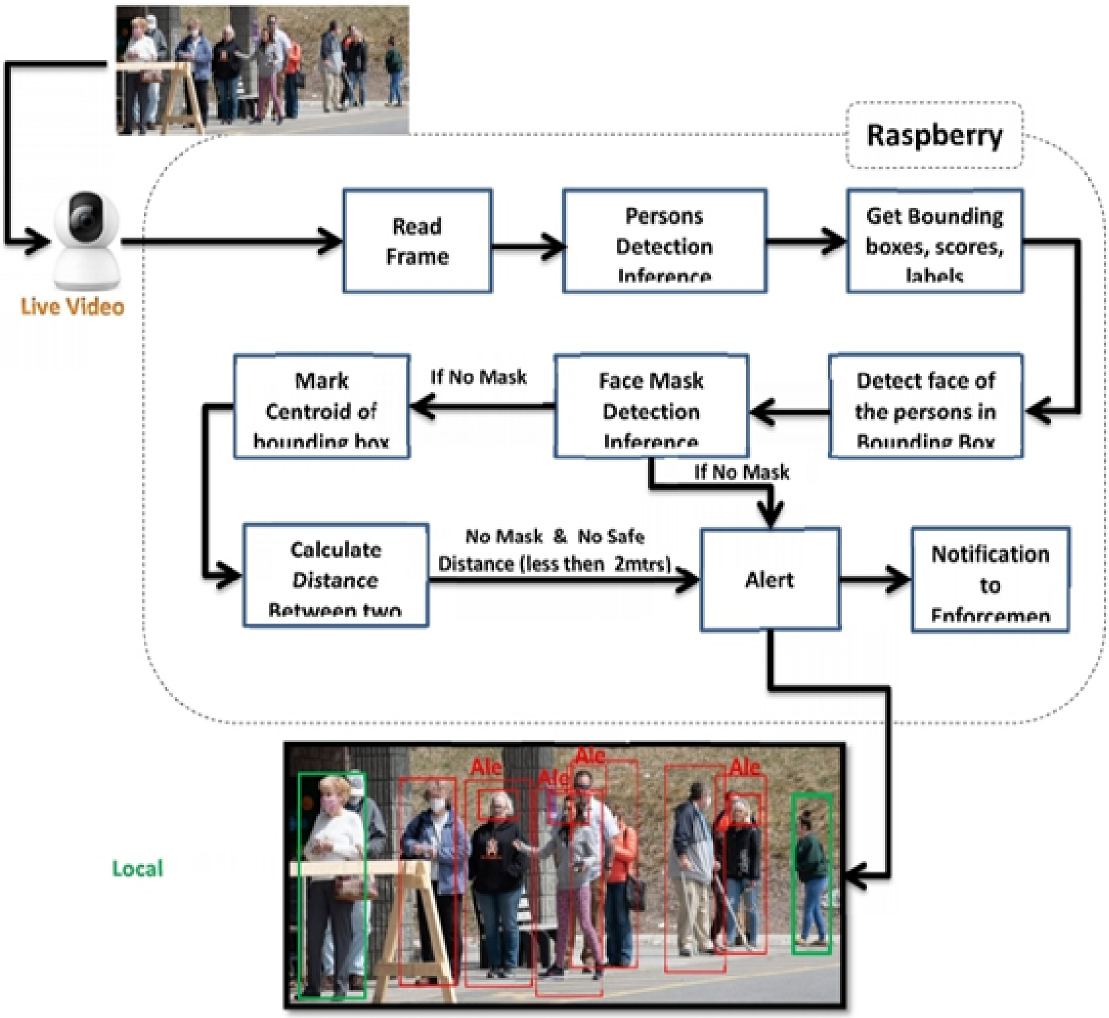


Fig 11 : Flowchart of Methodology

5.2 - Algorithms implemented

In Data Visualization we visualize the total number of images in our dataset that is there are 690 images in the with mask class and 686 images in the without_mask class. And after that we ***split*** our data into the ***training set*** which will contain the images on which the CNN model will be trained and the ***test set*** with the images on which our model will be tested.

OpenCV and Tensorflow have played a major role in our model for real time video processing. It is used to detect objects, faces, Face Mask etc. in real time video.

YOLOv3 was used for the Social distance module, YOLO stands for (You Only Look Once).It is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images.

In our Social Distance Monitoring model we have used YOLO to detect Persons in real time video.

Fig 12 : Libraries Used

5.3 - Datasets source and utilization

In this model, we're the usage of a face masks dataset created with the aid of using Prajna Bhandary. This dataset includes 1,376 pics belonging to masks and without masks 2 classes. Our foremost consciousness is to come across whether or not someone is carrying a masks or not, without getting near them. There was no Dataset used for Social distancing Monitoring, as we used YOLOv3 for calculating distance.

Fig 13 : Dataset Used

5.4 - Screenshots (GUI) of the project

- Here we can see the most important part of the code. In this code we have created a frame for face mask and then we integrated the code for social distancing code that we have extracted in the start. In the extracted file we had created a frame for social distancing and also added the counter for the number of people not following the social distance. Finally we integrated the face mask code and the social distancing code to run simultaneously and got the output as shown in the last.

In [1]:
```
# USAGE
# python social_distance_detector.py --input pedestrians.mp4
# python social_distance_detector.py --input pedestrians.mp4 --output output.avi

# import the necessary packages
from TheLazyCoder import social_distancing_config as config
from TheLazyCoder.detection import detect_people
from scipy.spatial import distance as dist
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import argparse
import imutils
import time
import cv2
import os
import sys
sys.argv=['']
del sys

def detect_and_predict_mask(frame, faceNet, maskNet):
    # grab the dimensions of the frame and then construct a blob
    # from it
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
        (104.0, 177.0, 123.0))

    # pass the blob through the network and obtain the face detections
    faceNet.setInput(blob)
    detections = faceNet.forward()
    print(detections.shape)

    # initialize our list of faces, their corresponding locations,
    # and the list of predictions from our face mask network
    faces = []
    locs = []
    preds = []

    # loop over the detections
```



```
    for i in range(0, detections.shape[2]):
        # extract the confidence (i.e., probability) associated with
        # the detection
        confidence = detections[0, 0, i, 2]

        # filter out weak detections by ensuring the confidence is
        # greater than the minimum confidence
        if confidence > 0.5:
            # compute the (x, y)-coordinates of the bounding box for
            # the object
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")

            # ensure the bounding boxes fall within the dimensions of
            # the frame
            (startX, startY) = (max(0, startX), max(0, startY))
            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

            # extract the face ROI, convert it from BGR to RGB channel
            # ordering, resize it to 224x224, and preprocess it
            face = frame[startY:endY, startX:endX]
            face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
            face = cv2.resize(face, (224, 224))
            face = img_to_array(face)
            face = preprocess_input(face)

            # add the face and bounding boxes to their respective
            # lists
            faces.append(face)
            locs.append((startX, startY, endX, endY))

    # only make a predictions if at least one face was detected
    if len(faces) > 0:
        # for faster inference we'll make batch predictions on all
        # faces at the same time rather than one-by-one predictions
        # in the above `for` loop
        faces = np.array(faces, dtype="float32")
        preds = maskNet.predict(faces, batch_size=32)

    # return a 2-tuple of the face locations and their corresponding
    # locations
    return (locs, preds)
```

```
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", type=str, default="",
    help="path to (optional) input video file")
ap.add_argument("-o", "--output", type=str, default="",
    help="path to (optional) output video file")
ap.add_argument("-d", "--display", type=int, default=1,
    help="whether or not output frame should be displayed")
args = vars(ap.parse_args())


# construct the argument parser and parse the arguments

# load the COCO class labels our YOLO model was trained on
labelsPath = os.path.sep.join([config.MODEL_PATH, "coco.names"])
LABELS = open(labelsPath).read().strip().split("\n")

# derive the paths to the YOLO weights and model configuration
weightsPath = os.path.sep.join([config.MODEL_PATH, "yolov3.weights"])
configPath = os.path.sep.join([config.MODEL_PATH, "yolov3.cfg"])

# load our YOLO object detector trained on COCO dataset (80 classes)
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

# check if we are going to use GPU
if config.USE_GPU:
    # set CUDA as the preferable backend and target
    print("[INFO] setting preferable backend and target to CUDA...")
    net.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
    net.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)

# determine only the output layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

# initialize the video stream and pointer to output video file
print("[INFO] accessing video stream...")
vs = cv2.VideoCapture(args["input"] if args["input"] else 0)
writer = None
```



```
# load our serialized face detector model from disk
print("[INFO] loading face detector model...")
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk
print("[INFO] loading face mask detector model...")
maskNet = load_model("mask_detector.model")



# loop over the frames from the video stream
while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break

    # resize the frame and then detect people (and only people) in it
    frame = imutils.resize(frame, width=700)
    results = detect_people(frame, net, ln,
        personIdx=LABELS.index("person"))

    # initialize the set of indexes that violate the minimum social
    # distance
    violate = set()

    # ensure there are at least two people detections (required in
    # order to compute our pairwise distance maps)
    if len(results) >= 2:
        # extract all centroids from the results and compute the
        # Euclidean distances between all pairs of the centroids
        centroids = np.array([r[2] for r in results])
        D = dist.cdist(centroids, centroids, metric="euclidean")

        # loop over the upper triangular of the distance matrix
        for i in range(0, D.shape[0]):
```

```
            for j in range(i + 1, D.shape[1]):
                # check to see if the distance between any two
                # centroid pairs is less than the configured number
                # of pixels
                if D[i, j] < config.MIN_DISTANCE:
                    # update our violation set with the indexes of
                    # the centroid pairs
                    violate.add(i)
                    violate.add(j)

    # loop over the results
    for (i, (prob, bbox, centroid)) in enumerate(results):
        # extract the bounding box and centroid coordinates, then
        # initialize the color of the annotation
        (startX, startY, endX, endY) = bbox
        (cX, cY) = centroid
        color = (0, 255, 0)

        # if the index pair exists within the violation set, then
        # update the color
        if i in violate:
            color = (0, 0, 255)

        # draw (1) a bounding box around the person and (2) the
        # centroid coordinates of the person,
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
        cv2.circle(frame, (cX, cY), 5, color, 1)

    # draw the total number of social distancing violations on the
    # output frame
    text = "Social Distancing Violations: {}".format(len(violate))
    cv2.putText(frame, text, (10, frame.shape[0] - 25),
        cv2.FONT_HERSHEY_SIMPLEX, 0.85, (0, 0, 255), 3)

    # check to see if the output frame should be displayed to our
    # screen
    #if args["display"] > 0:
        # show the output frame
        #cv2.imshow("Frame", frame)
        #key = cv2.waitKey(1) & 0xFF

        # if the `q` key was pressed, break from the loop
```



```
        #if key == ord("q"):
            #break

    # if an output video file path has been supplied and the video
    # writer has not been initialized, do so now
    if args["output"] != "" and writer is None:
        # initialize our video writer
        fourcc = cv2.VideoWriter_fourcc(*"MJPG")
        writer = cv2.VideoWriter(args["output"], fourcc, 25,
            (frame.shape[1], frame.shape[0]), True)

    # if the video writer is not None, write the frame to the output
    # video file
    if writer is not None:
        writer.write(frame)

# detect faces in the frame and determine if they are wearing a
    # face mask or not
    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

    # loop over the detected face locations and their corresponding
    # locations
    for (box, pred) in zip(locs, preds):
        # unpack the bounding box and predictions
        (startX, startY, endX, endY) = box
        (mask, withoutMask) = pred

        # determine the class label and color we'll use to draw
        # the bounding box and text
        label = "Mask" if mask > withoutMask else "No Mask"


        color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

        # include the probability in the label
        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

        # display the label and bounding box rectangle on the output
        # frame
        cv2.putText(frame, label, (startX, startY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
```

5/9/2021 final - Jupyter Notebook

```
    # show the output frame
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break



# do a bit of cleanup
cv2.destroyAllWindows()
```

```
[INFO] loading YOLO from disk...
[INFO] accessing video stream...
[INFO] loading face detector model...
[INFO] loading face mask detector model...
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
```

Fig 14 : Screenshot of the Code

Output :- The output obtained from the code is shown below:-

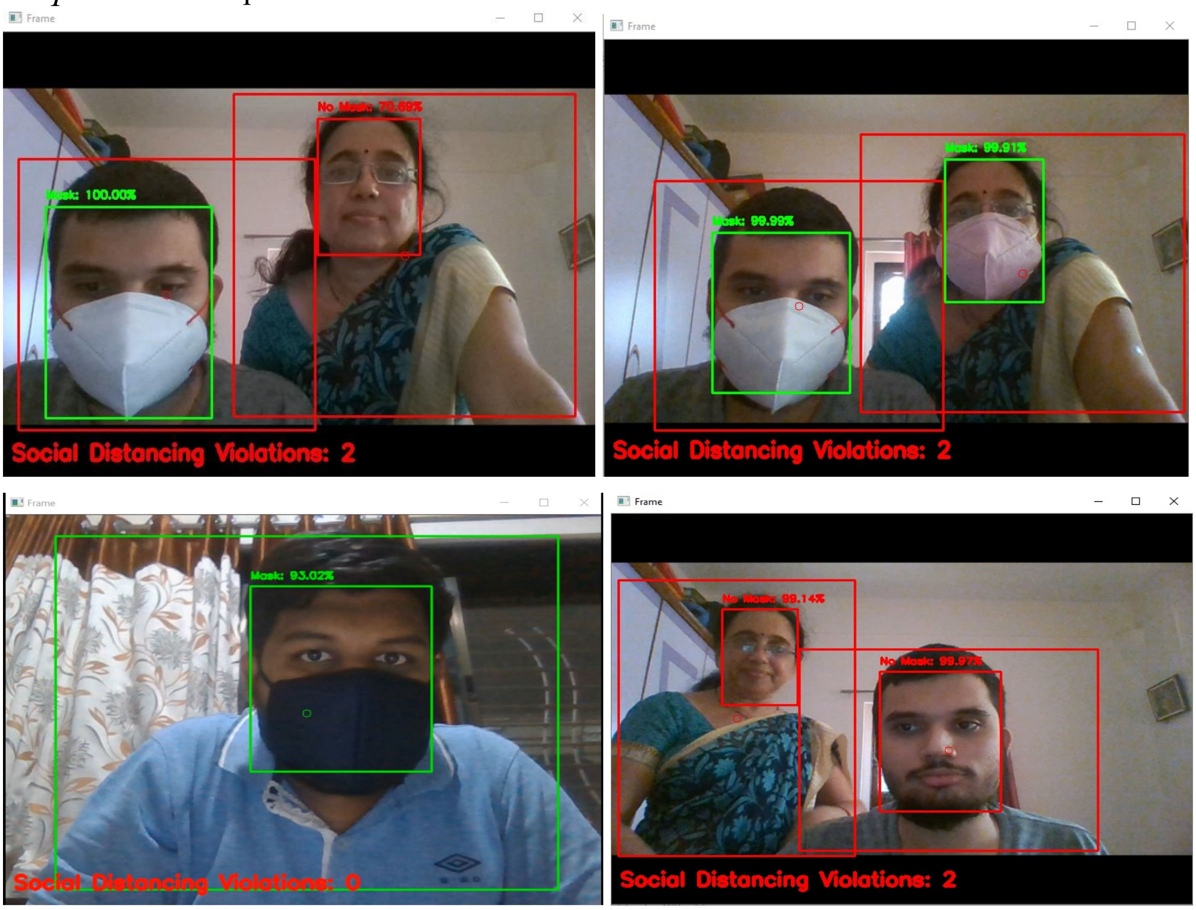
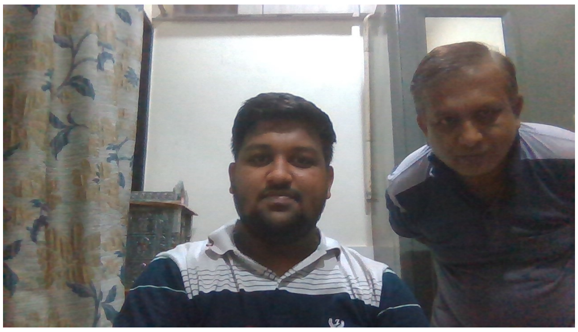


Fig 15 : Output obtained

Chapter 6 - Testing of the Proposed System

After implementing the code now let us look at some of the test cases we tried and what were the results we obtained. Every output obtained is explained below.

6.1 Test Cases

	Test Cases
Case 1] *Model check on a video* :- In this we are testing our model on a video whether it can predict correctly or not.	
Case 2] *Model check on a real time facemask* :- In the second test case we are checking whether our face mask model is running good for real time.	
Case 3] *Model check for real time social distance* :- In the third test case we are checking whether our model is able to detect the social distance .	
Case 4] *Model check for real time face mask and social distance simultaneously* :- This is our 4th and final test case where we are checking the social distance and face mask simultaneously and see the accuracy.	

<u>## 6.2 - Test Result</u>

For Case 1]

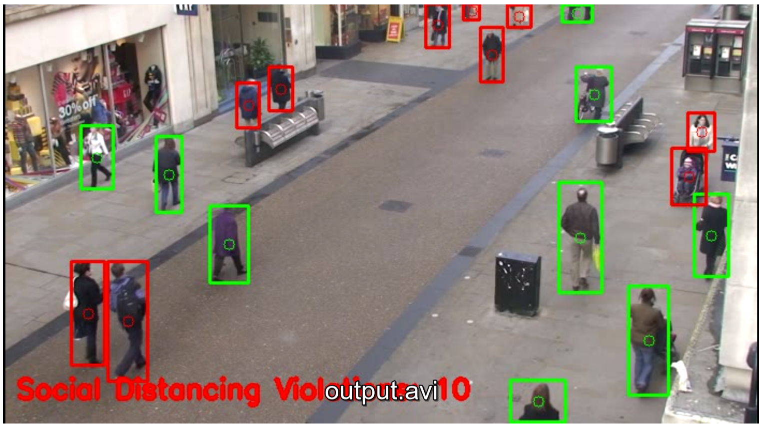


Fig 16 : Result of Test case 1

As it can be seen from the image above the model is running fine on a video and showing us the counter for those who are not maintaining social distance. The red boxes are the once that are breaking the rules and the green ones are the ones that are safe.

For Case 2]

For test case 2 there were 3 results obtained. Let us look at them one by one:-

a] When the person is not wearing a mask then it is shown in a red frame and with a No mask percent of almost 100% as shown below.

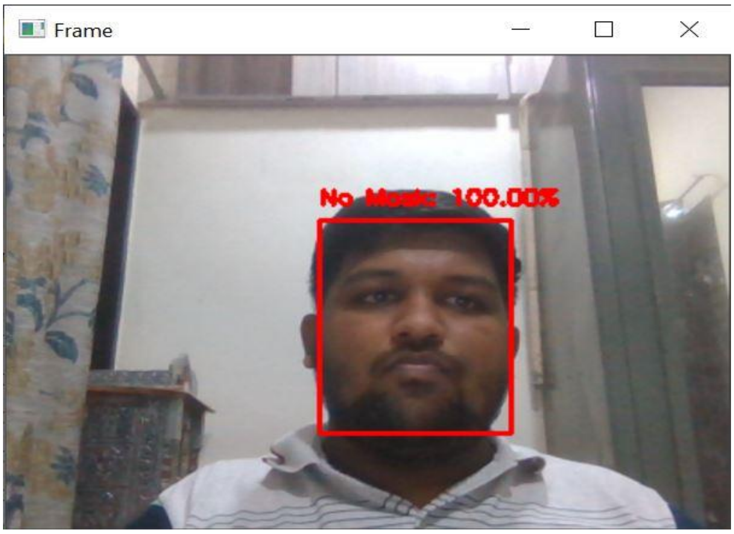


Fig 17 : Result A of Test case 2

b] When the person is wearing a mask completely then it is shown in a green frame and either a mask percent of almost 100% as shown below.

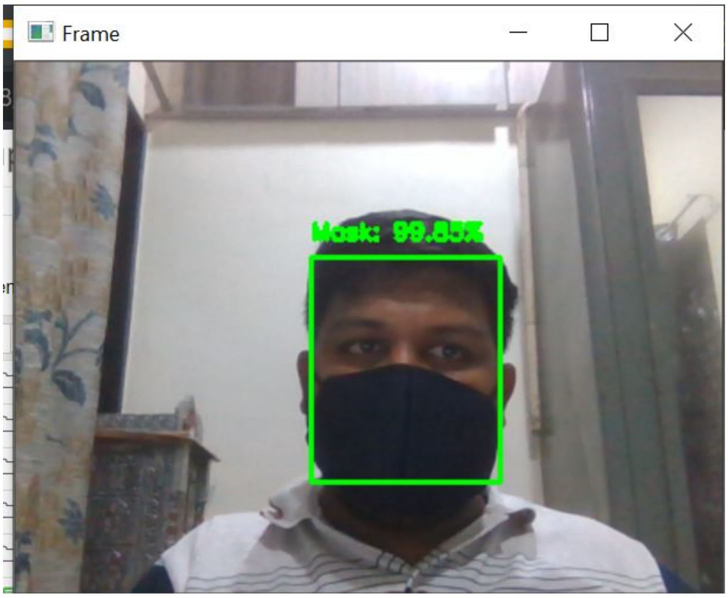


Fig 18 : Result B of Test case 2

c] When the person is wearing a mask partially then it can show a red or green frame depending on the position of the mask and accordingly the mask or no mask percent will be displayed as shown below.

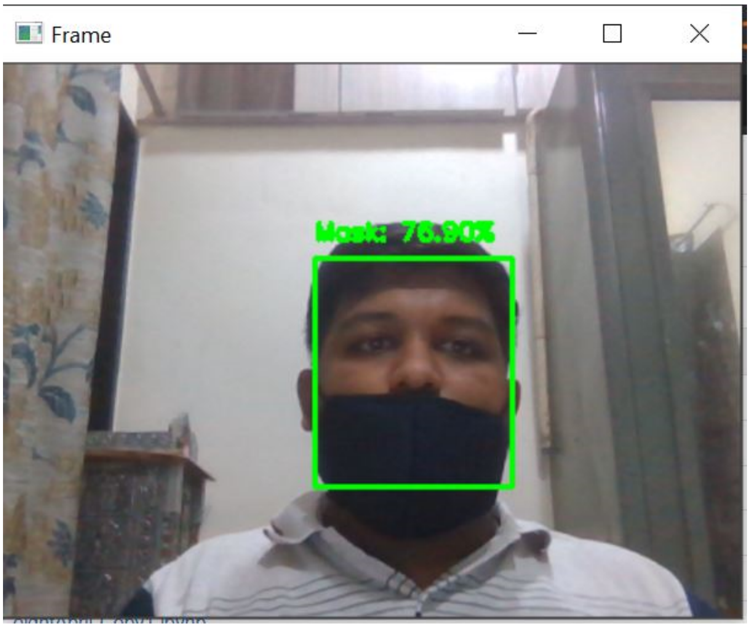


Fig 19 : Result C of Test case 2

For Case 3]

For case 3 we have 2 results. Let us see the outcome one by one :-

a] When a single person is in the frame so a green box will be shown saying that the person is safe and the counter will be 0 as shown in the image below.

Fig 20 : Result A of Test case 3

b] When more than one person is in the frame and if they are not maintaining 5000 pixel(2m) distance then a red box will be shown on everyone of them and the counter will increase accordingly for the number of people not following social distancing as shown in the image.

Fig 21 : Result B of Test case 3

For Case 4]

For our last test case there are a total 5 results obtained. Let's find out the outcomes one by one :-

a] When a person is alone in the frame and is not wearing a mask then the outer frame will be green indicating that he/she is maintaining social distancing but the inner frame will show red indicating that the person is not wearing a mask as shown in the image.

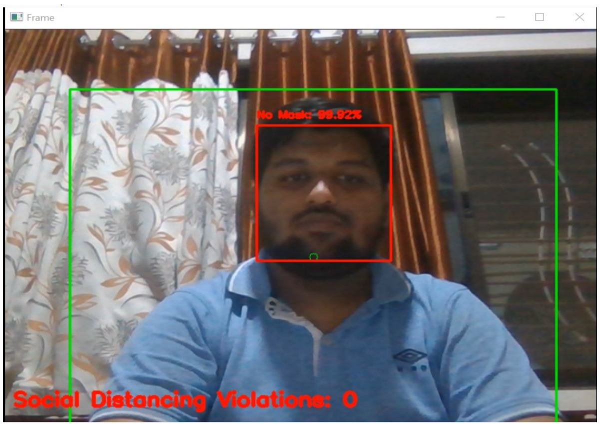


Fig 22 : Result A of Test case 4

b] When a person is alone in the frame and is wearing a mask then the outer frame will be green indicating that he/she is maintaining social distancing and the inner frame will also show green indicating that the person is wearing a mask as shown in the image.

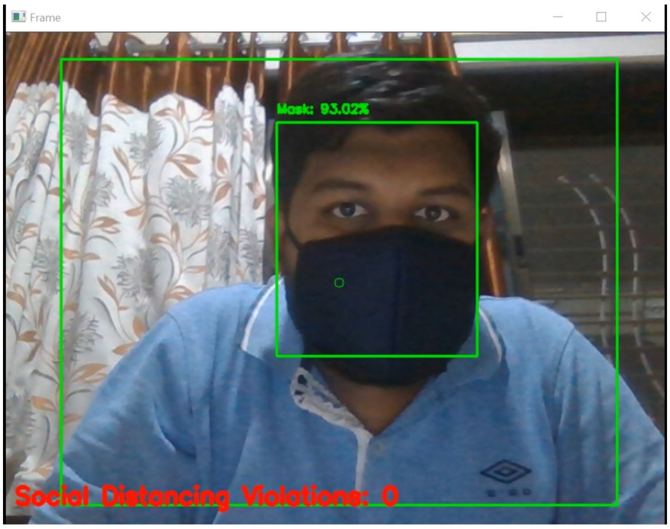


Fig 23 : Result B of Test case 4

c] When there are more than 1 person in the frame and if they are not maintaining social distance as well as they are not wearing a mask then outer frame as well as inner frame will be red indicating that neither they are following social distancing nor they are wearing mask and also the counter will increase as shown below.

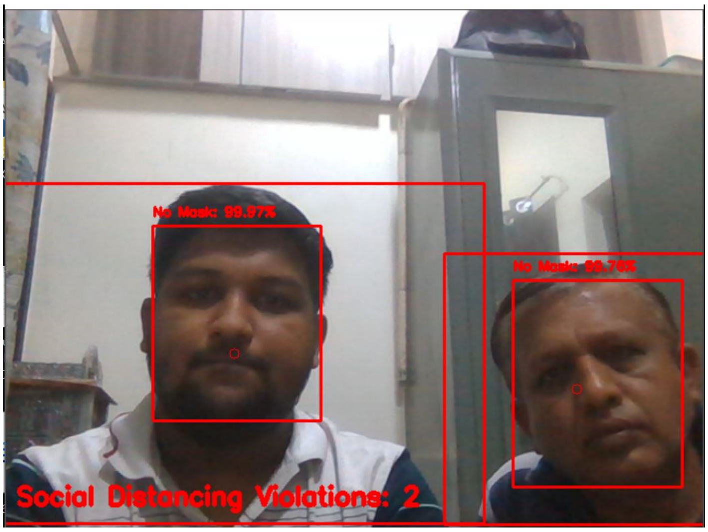


Fig 24 : Result C of Test case 4

d] When there are more than 1 person in the frame and if they are not maintaining social distance but one of them is wearing a mask then outer frame will be red indicating they are not maintaining social distance and one inner frame will be red showing the person is not wearing a mask and another inner frame will be green indicating that the person is wearing a mask properly but the counter will increase as shown below.

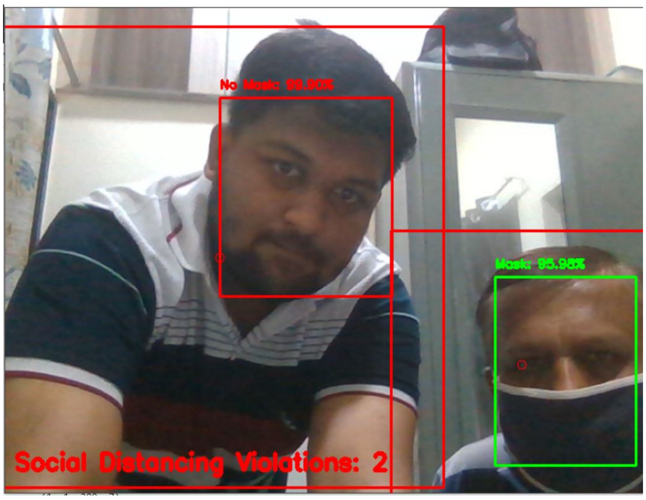


Fig 25 : Result D of Test case 4

e] When there are more than 1 person in the frame and if they are not maintaining social distance but all of them are wearing a mask then outer frame will be red indicating they are not maintaining social distance but the inner frame will be green showing that all are wearing mask but the counter will increase as they are not maintaining social distance as shown below.

Fig 26 : Result E of Test case 4

Chapter 7 - Result Analysis

In this Chapter we have discussed the final result we obtained and compared the accuracy score from the research we have done. You would be amazed by the accuracy we got.

7.1 - Performance Evaluation measures

The system is a deep learning solution that uses OpenCV and TensorFlow, to train the model. We combine the deep learning YOLOv3 module with the SSD framework for a fast and efficient deep learning solution for real-time human detection in video streams and use a triangular similarity technique to measure distance between persons detected by camera in real time in public places and comprises customized data collection to resolve a face mask detection model with variance in the types of face masks worn by the public in real time by means of a transfer of learning to a pre-trained SSD face detector. This model combine's social distance detection and face mask detection.
In the proposed system, three steps are followed, such as:
1) Model development and training
2) Model testing
3) Model implementation
A] Model development and training :- Our framework uses the transfer learning approach and will fine-tune the MobileNetV2 model, which is a highly efficient architecture that can be applied to edge devices with limited computing power, such as raspberry pi4 to detect people in real time. We used 80% of our total custom data set to train our model with a single shot detector, which takes only one shot to detect multiple objects that are present in an image using multibox. The custom data set is loaded into the project directory and the algorithm is trained on the basis of the labeled images. In pre-processing steps, the image is resized to 224×224 pixels, converted to numpy array format and the corresponding labels are added to the images in the dataset before using our SSD model as input to build our custom model with MobileNetV2 as the backbone and train our model using the TensorFlow Object Detection API. We also use the YOLOv3 model for calculating the distance between humans. It creates a frame and objects and using k means it finds the distance between the objects.

B] Model testing :- The 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 sends an alert to monitoring authorities with a face image. The system detects the social distancing and masks with a precision score of 91.7% .

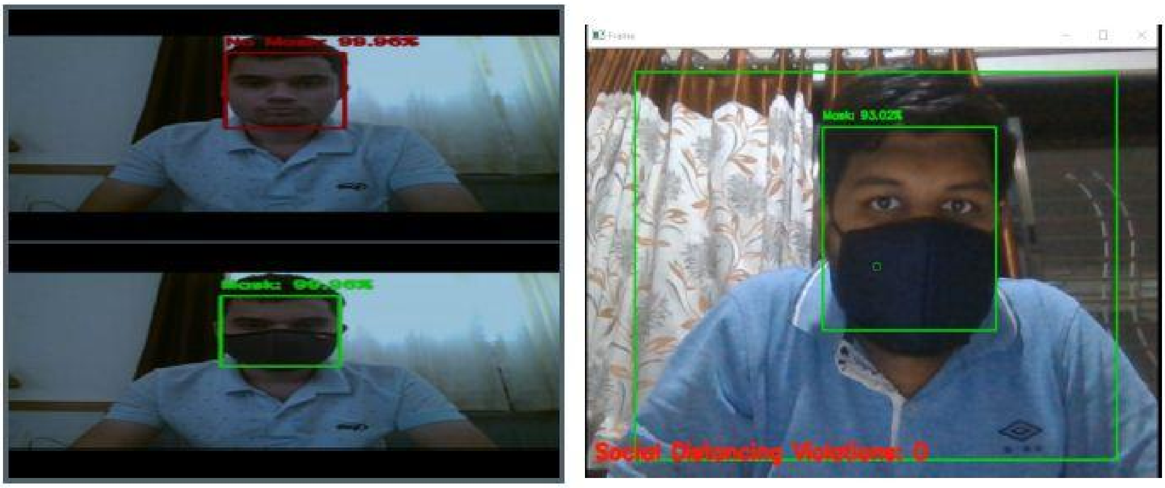


Fig 27 : Model testing Result

C] Model implementation :- The system uses raspberry pi4 with a camera to automatically track public spaces in real-time to prevent the spread of Covid-19. The trained model with the custom data set is installed in the raspberry pi4, and the camera is attached to it. The camera feeds real-time videos of public places to the model in the raspberry pi4, which continuously and automatically monitors public places and detects whether people keep safe social distances and also checks whether or not those people wear masks. When the detection of a social distance violation by individuals is detected continuously in threshold time, there will be an red alert that instructs people to maintain social distance.

Fig 28 : Model implementation Result

7.2 - Input Parameters / Features considered

Step 1:Train a face mask detection with mass images

The first step is to input over 200,000 masked face photos and 1,000,000 of the standard face images into an in-depth learning dataset. Usually, the mask recognition deviation rate will be just 5-10% after the first round comprehensive learning training.

Step 2:Retrain with the rejected image

To raise the mask recognition deviation rate, repeat step one; this time ,only upload incorrectly recognized photos into the deep learning network until the mask recognition accuracy reaches at least 99.98% in the lab environment.

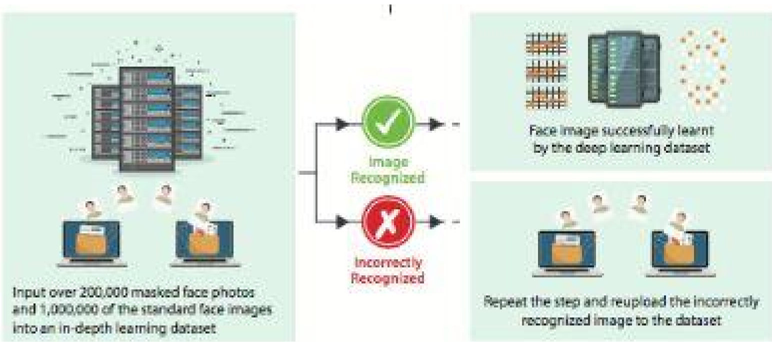


Fig 29 : Retrain with the rejected image

Step 3: Image differentiate

Once the face mask detector is well-trained, we can then move on to loading the mask detector, starting its face function, and classifying the image as ”With mask” or “Without a mask”. With the field application, the lowest acceptable standard will be atleast 98% accuracy under 10,000 field test.

Fig 30 : Image Differentiate

7.3 - Graphical and statistical output

- ❖ As we can see below a graphical representation of Accuracy. It is clearly having an accuracy score of around 99.44%. And this we got after running 20 epochs. The training loss was also very low as the epoche increased.

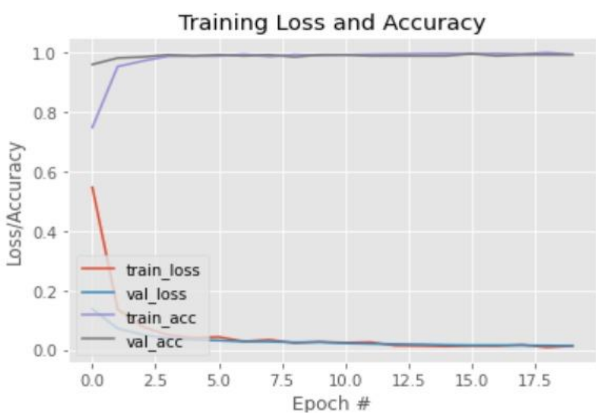


Fig 31 : Accuracy and Training loss Graph

7.4 - Comparison of results with existing systems

- As we discussed in the literature survey above the accuracy score was around 75 - 95 % . According to the authors their models were able to predict in a good manner.

- According to our model the accuracy score is about 99.44% which is quite high compared to the exciting model of the above authors.

	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	138
without_mask	0.99	0.99	0.99	138
accuracy			0.99	276
macro avg	0.99	0.99	0.99	276
weighted avg	0.99	0.99	0.99	276

Fig 32 : Accuracy Score

7.5 - Inference drawn

We implemented our model on images containing one and more faces. We also implemented it on videos and live video streams by removing and wearing masks one by one. Some screenshots of the results are shown below:

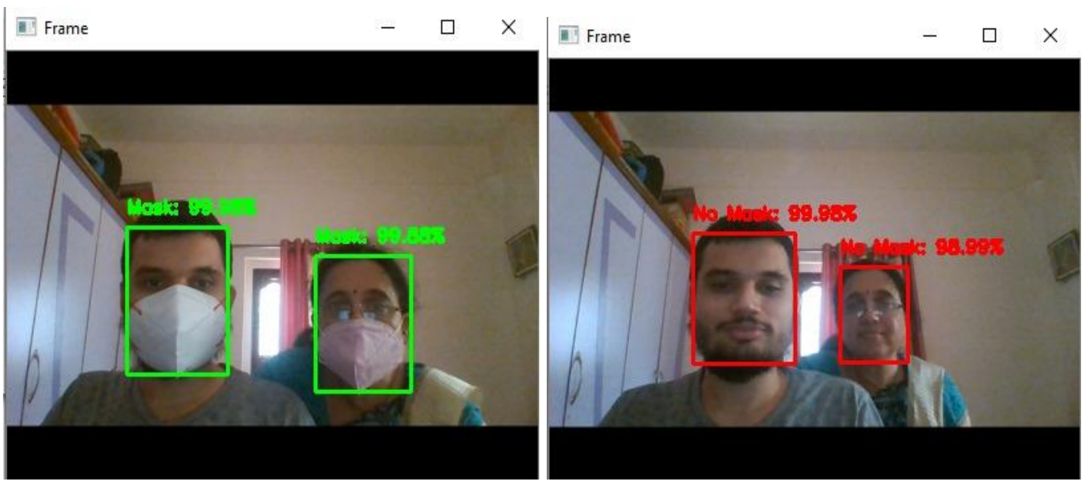


Fig 33 : Inference Drawn

Chapter 8 - Conclusion

This is the final chapter where we are concluding our topic discussing some of the limitations and the future scope.

8.1 - Limitations

There were not many challenges faced but the two problems that were time consuming and made the tasks tedious are discussed as follows. One was the excessive data loading time in Google ColabNotebook while loading the dataset into it. Since the runtime restarting refreshes all the cells, the cell for dataset loading took most of the time while running. Secondly, the access problem in GoogleColab Notebook: it did not allow the access of webcam which posed a hurdle in testing images and live video stream through Google Colab Notebook. Therefore, we had to run the code locally on the computer through which we tested the code on the live video stream.

8.2 - Conclusion

In this project, we construct a version which could stumble on actual time face mask and additionally assist in tracking the social distancing on this pandemic scenario. As noted above we've used many libraries and attempted many algorithms. Modules like YOLOv3 and tensorflow had been a number of the maximum vital libraries of our version. It will assist hold a stable surroundings and make certain people safety by means of robotically tracking public locations to keep away from the unfold of the COVID-19 virus through digital digicam feeds with raspberry pi4 in actual-time. Thus, this gadget will function in an green way withinside the modern scenario whilst the lockout is eased and facilitates to tune public locations without difficulty in an automatic way. We have addressed intensive the monitoring of social distancing and the identity of face mask that assist to make certain human health.. The answer has the capacity to noticeably lessen violations by means of actual-time interventions, so the proposed gadget could enhance public protection via saving time and assisting to lessen the unfold of coronavirus. This answer may be utilized in locations like schools, colleges, temples, purchasing complexes, metro stations, airports, etc.

8.3 - Future Scope

1) Coughing and Sneezing Detection: Chronic coughing and sneezing is one of the key signs and symptoms of COVID-19 contamination as according to WHO suggestions and additionally one of the main routes of ailment unfold to non-inflamed public. Deep mastering primarily based totally technique may be proved reachable right here to detect & restrict the ailment unfold through improving our proposed answer with frame gesture evaluation to recognize if an person is coughing and sneezing in public places whilst breaching facial mask and social distancing tips and based mostly on very last outcomes enforcement businesses can be alerted.

2) Temperature Screening: Elevated frame temperature is any other key symptom of COVID-19 infection, at gift situation thermal screening is performed the usage of hand held contactless IR thermometers wherein medical examiner want to are available in near proximity with the man or woman want to be screened which makes the medical examiners susceptible to get inflamed and additionally its nearly not possible to seize temperature for every and each person in public locations, the proposed use-case may be ready with thermal cameras primarily based totally screening to research frame temperature of the peoples in public locations which could upload any other assisting hand to enforcement businesses to address the pandemic effectively.

References

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol.1. IEEE, 2001, pp. I–I.

[2] Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning by Adrian Rosebrock

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition,2014, pp. 580–587.

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[7] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015,pp. 1440–1448.

[8] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015,pp. 1440–1448.

[9] Enabling and Emerging Technologies for Social Distancing: A Comprehensive Survey

[10] Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence.

[11] Applying deep learning algorithms to maintain social distance in public place through drone technology.

Patents reference:

https://patentscope.wipo.int/search/en/result.jsf?_vid=P10-KOLQ2F-99868

Appendix

1] Paper Details

a] Paper Published

12th ICCCNT 2021 Submission 362

If you want to **change any information** about your paper, use links in the upper right corner.

For all questions related to processing your submission you should contact the conference organizers. Click here to see information about this conference.

Submission 362	
Title:	Real Time Face Mask Detector and Social Distance Monitoring
Paper:	(May 09, 14:00 GMT)
Author keywords:	Machine Learning Algorithm Tensorflow OpenCV numpy sklearn Argparser Imutils YOLOv3
EasyChair keyphrases:	social distance (140), social distancing (130), face mask (115), public location (110), digital digicam (80), actual time (80), vivekanand education society (79), education society institute (79), cmpn vivekanand education (79), real time (70), getting to know (63), face mask detection (63), computer vision (60), technology mumbai (50), coughing and sneezing (47), actual time face mask (40), bounding box (40), real time object detection (40)
Abstract:	The Objective of our assignment is to hold Social Distance amongst humans and to test Face Mask on face of peoples withinside the time of COVID-19. This version may be detecting actual time Face masks and Social Distancing which may be very vital on this pandemic situation. Since all of the schools, schools and places of work which are closed now will reopen quickly and they may be wanting a few generations to be secure and healthy.
Submitted:	May 09, 14:00 GMT
Last update:	May 09, 14:02 GMT

Authors						
first name	last name	email	country	affiliation	Web page	corresponding?
Yashkumar	Jain	yashkumarjain1200@gmail.com	India	Vivekanand Education Society Institute Technology		✓
Shreyas	Kotkar	2018.shreyas.kotkar@ves.ac.in	India	Vivekanand Education Society Institute Technology		✓
Vikram	Virwani	2018.vikram.virwani@ves.ac.in	India	Vivekanand Education Society Institute Technology		✓
Chirag	Kinger	2018.chirag.kinger@ves.ac.in	India	Vivekanand Education Society Institute Technology		✓
Sujata	Khedkar	sujata.Khedkar@ves.ac.in	India	Vivekanand Education Society Institute Technology		✓

Fig 34 : Paper Published

Submission 63	
Title:	Real Time Face Mask Detection and Social Distance Monitoring
Paper:	(May 10, 18:00 GMT)
Author keywords:	Tensorflow OpenCV numpy YOLOv3
Abstract:	The Objective of our assignment is to hold Social Distance amongst humans and to test Face Mask on face of peoples with inside the time of COVID-19 . This version may be detecting actual time Face masks and Social Distancing which may be very vital on this pandemic situation. Since all of the schools, schools and places of work which are closed now will reopen quickly and they may be wanting a few generations to be secure and healthy.
Submitted:	May 10, 18:00 GMT
Last update:	May 10, 18:00 GMT

Authors						
first name	last name	email	country	affiliation	Web page	corresponding?
Yashkumar	Jain	yashkumarjain1200@gmail.com	India	Vivekanand Education Society Institute Of Technology		✓
Shreyas	Kotkar	2018.shreyas.kotkar@ves.ac.in	India	Vivekanand Education Society Institute Technology		✓
Vikram	Virwani	2018.vikram.virwani@ves.ac.in	India	Vivekanand Education Society Institute Of Technology		✓
Chirag	Kinger	2018.chirag.kinger@ves.ac.in	India	Vivekanand Education Society Institute Of Technology		✓
Sujata	Khedkar	sujata.khedkar@ves.ac.in	India	Vivekanand Education Society Institute Of Technology		✓

Fig 35 : Paper Published

Paper Link :-
https://drive.google.com/file/d/1Nwa4MFuckloyqefDfitj9Rhgzbpr6Hyd/view?usp=sharing

b] Plagiarism Report

Plagiarism Scan Result Report

Scanned Text

Input File	*Copy of Real Time Face Mask Detection and Social Distance Monitoring (1).pdf*
Author	*n/a*
Scan Date	*2021-05-07*
Project	*Sonstige Tests*
Remarks	*n/a*

Result Overview

Copied/Total Words: 81/2475
Suspicious Sources: 1
Accepted Sources: 0
Excluded Sources: 0

Percentage of Copied Words / Total Words

3.27%

Overview of Matching Phrases

Fig 36 : Plagiarism Report(score)

Report link:-
https://drive.google.com/file/d/1BST8CURNPw5cgJi2HsB4nO627t7FI1yh/view?usp=sharing

c] Project Review Sheet

Inhouse/ Industry:
Mentor Name:
Group No.:

Project Evaluation Sheet 2020 - 21

Class: D12 A/B/C
Group No.:

Title of Project: Real time Face Mask Detection and Social Distance Monitoring System

Group Members: Chirag Kinger 31/D12C, Yashkumar Jain 24/D12B, Shreyas Kotkar 34/D12C, Vikram Virwani 68/D12B.

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Research & Innovative Approach (5)	**Total Marks (50)**
4	4	5	3	5	2	2	2	2	3	3	3	4	5	47

Comments: Very Nice Work

Dr. Sujata Khedkar
Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Research & Innovative Approach (5)	**Total Marks (50)**
4	4	5	3	5	2	2	2	2	3	3	3	4	5	47

Comments:

Date: 17th March,2021

Name & Signature Reviewer2
Richa Sharma

Fig 37 : Review Sheet 1

Inhouse/ Industry:
Mentor Name:
Group No.:

Project Evaluation Sheet 2020 - 21

Class: D12 A/B/C
Group No.:

Title of Project: Real time Face Mask Detection and Social Distance Monitoring System

Group Members: Chirag Kinger 31/D12C, Yashkumar Jain 24/D12B, Shreyas Kotkar 34/D12C, Vikram Virwani 68/D12B.

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Research & Innovative Approach (5)	Total Marks (50)
4	4	4	3	5	2	2	2	2	3	3	3	4	5	46

Comments:

Dr. Sujata Khedkar

Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Research & Innovative Approach (5)	Total Marks (50)
4	4	4	3	5	2	2	2	2	3	3	3	4	5	46

Comments:

Date: 19th April,2021

Name & Signature Reviewer2

Richa Sharma

Fig 38 : Review Sheet 2

2] Video Link :- https://youtu.be/sFxxMk4IDrs