

COVID - 19 PREVENTION MELIORATION: FACE MASK & SOCIAL DISTANCING DETECTION

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

by

Aishwarya.S

(Reg. No. 37110025)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

March - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC | Approved by AICTE
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Ms. Aishwarya.S (37110025)** who carried out the project entitled " **COVID - 19 PREVENTION MELIORATION: FACE MASK & SOCIAL DISTANCING DETECTION** " under my supervision from November 2020 to March 2021.

Internal Guide
Dr. V. NIRMALRANI

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **Aishwarya.S (37110025)** hereby declare that the Project Report entitled "**COVID - 19 PREVENTION MELIORATION: FACE MASK & SOCIAL DISTANCING DETECTION**" done by me under the guidance of **Dr.V.Nirmalrani** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D .**, Dean, School of Computing and **Dr.L.Lakshmanan M.E., Ph.D.,and Dr.S.Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.V.Nirmalrani** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The rampant Coronavirus disease 2019 (COVID-19) has brought global crisis with its deadly spread to more than 180 countries, and about 3,519,901 confirmed cases along with 247,630 deaths globally as on May 4, 2020. Now that the vaccination has been found, WHO has still insisted on wearing masks to protect against this deadly virus and maintaining social distancing as it is the most feasible approach to fight against this pandemic. Face detection and object detection has emerged as a very interesting problem in image processing and computer vision. It is more relevant today because it not only used on images but also in videos. Object detection using deep learning is helpful in computer vision. These detection algorithms works efficiently in current situation. This notion motivated me to contribute an efficient computer vision based detection system to public safety. The face mask detection system and social distancing detection system that exist works separately yet efficiently. In this proposed system I have brought both together, it is difficult yet new. The detection system detects if there is any violations through the camera and or on any video or image input. In this system, we have used deep learning algorithm. This system I have proposed here can be applied in places where people gather in huge amount. This automated detection technique makes it easier to inspect the public gathering in malls, streets, any firms, apartments, etc.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NUMBER
	ABSTRACT	i
	LIST OF FIGURES	iv
	LIST OF TABLES	vi
	LIST OF ABBREVIATION	vii
1.	INTRODUCTION	1
	1.1 GENERAL	1
	1.2 PROJECT OUTLINE	1
	1.3 OBJECTIVE	2
	1.4 MOTIVATION	3
2.	LITERATURE SURVEY	4
	2.1 RELATED WORK	4
3.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	9
	3.1 GOAL OF THE PROJECT	9
	3.2 PROBLEM STATEMENT	9
	3.3 SCOPE OF THE PROJECT	10
4.	EXPERIMENTS OR MATERIALS AND METHODS; ALGORITHMS USED	12
	4.1 SYSTEM DEVELOPMENT ENVIRONMENT	12
	4.2 SYSTEM REQUIREMENTS	16
	4.3 SOFTWARE REQUIREMENT SPECIFICATION	16
	4.4 MODULE IMPLEMENTATION	19
	4.5 FLOW OF THE PROPOSED SYSTEM	25
5.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	26

5.1 INTRODUCTION	26
5.2 REQUIREMENT ANALYSIS	26
5.3 SYSTEM ANALYSIS	28
5.4 SYSTEM DESIGN	30
5.5 SYSTEM EVOLUTION	31
5.6 RESULT	32
6. SUMMARY AND CONCLUSION	35
6.1 INTRODUCTION	35
6.2 SUMMARY	35
6.3 CONCLUSION	35
6.4 FUTURE WORK	36
REFERENCES	37
APPENDICIES	40
A. SOURCE CODE	40
B. SCREEN SHOTS	44
C. PUBLICATION WITH PLAGIARISM REPORT	47

LIST OF FIGURES

FIGURE NUMBER	FIGURE	PAGE NUMBER
1.1	OBJECT DETECTION EXAMPLE	1
3.1	GENERAL FLOW OF THE PROPOSED SYSTEM	11
4.1	COMPUTER VISION CONCEPTUAL DIAGRAM	13
4.2	DEEP LEARNING RELATIONSHIP DIAGRAM	13
4.3	IMAGE CLASSIFICATION EXAMPLE	14
4.4	OBJECT DETECTION EXAMPLE	15
4.5	FACE MASK DETECTION FLOW DIAGRAM	20
4.6	CAFFE MODEL ARCHITECTURE	21
4.7	POINT WISE AND DEPTH WISE CONVOLUTIONS	22
4.8	FACE MASK DETECTION MODEL ACCURACY	22
4.9	SOCIAL DISTANCE DETECTION FLOW DIAGRAM	23
4.10	YOU ONLY LOOK ONCE ARCHITECTURE	24
4.11	FLOW OF THE PROPOSED SYSTEM	25

5.1	ANALYSIS MODEL	30
5.2	GENERAL FLOW OF THE SYSTEM	30
5.3	PEOPLE WEARING FACE MASK AND VIOLATING SOCIAL DISTANCEIN AN IMAGE	32
5.4	DETECTION OF VIOLATION IN A VIDEO INPUT	33
5.5	VIOLATION DETECTION ON A LIVE CAMERA	34

LIST OF TABLES

TABLE NUMBER	TABLE	PAGE NUMBER
4.1	HARDWARE REQUIREMENTS	16
4.2	SOFTWARE REQUIREMENTS	16

LIST OF ABBREVIATIONS

COVID	Corona Virus Disease
WHO	World Health Organization
CNN	Convolutional Neural Network
YOLO	You Only Look Once
EOD	Explosive Ordnance Disposal
SSD	Single Shot multi-box Detector
F-RCNN	Faster Region-based Convolutional Neural Network
GUI	Graphical User Interface
CV	Computer vision
SRS	Software Requirement Specification
OPENCV	Open Source Computer Vision Library

API	Application Programming Interface
ML	Machine Learning
ROI	Region Of Interest
CPU	Central Processing Unit
GPU	Graphical Processing Unit
RAM	Random Access Memory
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
R-CNN	Region Based Convolutional Neural Networks

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Computer vision is the most powerful and compelling types of Artificial Intelligence. The development of computer vision has made its mark recently in analyzing and solving the complexive problems which has the relevant potential to contribute and help the health care industry in controlling COVID-19.

In recent years, deep learning techniques are achieving state-of-the-art results for object detection, such as on standard benchmark datasets and in computer vision competitions. Deep learning techniques are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN). Notable is the “You Only Look Once,” or YOLO, family of Convolutional Neural Networks that achieve near state-of-the-art results with a single end-to-end model that can perform object detection in real-time.

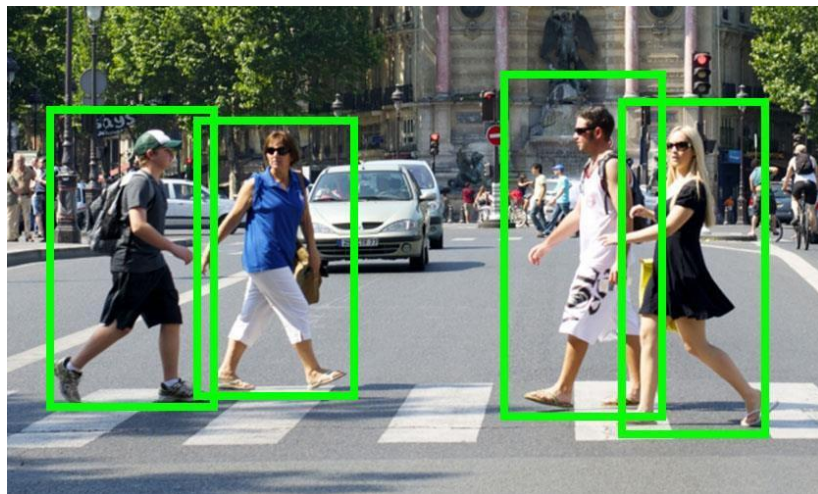


Fig 1.1: Object Detection Example

1.2 PROJECT OUTLINE

Coronavirus disease 2019 has affected the world seriously. Major protection method for people is to wear masks in public areas and maintain social distancing. Furthermore, many public service providers require customers to use the service only

if they wear masks correctly and follow social distancing. However, there are only a few research studies about face mask detection and social distancing together.

Respiratory droplets are generated when an infected person coughs or sneezes. Any person in close contact (within 1 m) with someone who has respiratory symptoms (coughing, sneezing) is at risk of being exposed to potentially infective respiratory droplets. Thus, wearing mask and maintaining distance is essential.

Droplets may also land on surfaces where the virus could remain viable; thus, the immediate environment of an infected individual can serve as a source of transmission (contact transmission). Wearing a medical mask and maintaining social distancing are the prevention measures that can limit the spread of certain respiratory viral diseases, including COVID-19.

In this project, I have proposed a computer vision based system which detects both presence of face mask on people and social distance between people. The system takes input from a live video, images and videos. The system is developed with the combination of various architectures. The ResNet is used for face detection, MobileNetV2 is used as backbone for feature extraction to achieve a best face mask detection model and YOLO is used for object detection. Using the above methods, the system detects the face then looks for the presence of face mask and also compares the distance between people simultaneously. The system thus makes it easier to detect people who are violating the measures in single point of control.

1.3 OBJECTIVE

Main objectives of this proposed system is to design a system to prevent people from infections. This can be done by identifying the person on image, video or live stream from a camera who is not wearing face mask and not maintaining social distance. It can be detected with the help of computer vision and deep learning algorithm. Also, to provide top management in a single point of control that will be

helpful to reduce violations and decrease the spreading of infection.

1.4 MOTIVATION

One of the increasing popular public concerns is human health. In the present scenario due to Covid-19, there is no efficient face mask and social distancing detection applications together which are now in high demand for transportation means, densely populated areas, residential districts, large-scale manufacturers and other enterprises to ensure safety. If early actions can be taken in time, then lots of people can be saved. However, face mask detection and social distancing detection separately is inconvenient, time consuming and expensive. Therefore, these constraints motivated me to provide all in single slate for detecting both face mask and social distance, so people have the advantage of easy and simple access.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

The face mask and social distancing are the best technique to prevent from the corona virus infection, initially when COVID-19 occurred in end of 2019 the above mentioned ways was brought as an effective measure and it is still has positive impact. There are various papers and researches related to the face mask detection and social distancing detection. I have studied the existing papers and learned about the advantages and disadvantages of the existing work. Therefore, by learning the advantages I have applied those in my proposed system and by analysing the disadvantages I have managed to come up with feasible and efficient solutions for the problems in the existing system.

2.1.1 Object Detection Using Convolutional Neural Networks

Reagan L. Galvez; Argel A. Bandala; Elmer P. Dadios; Ryan Rhay P. Vicerra and Jose Martin Z. Maningo have put forward the concept of object detection using convolutional neural networks in 2019. They stated that vision systems are essential in building a mobile robot that will complete a certain task like navigation, surveillance, and explosive ordnance disposal (EOD). That will make the robot controller or the operator aware what is in the environment and perform the next tasks. With the recent advancement in deep neural networks in image processing, classifying and detecting the object accurately is now possible. In their paper, Convolutional Neural Networks (CNN) is used to detect objects in the environment. Two state of the art models are compared for object detection, Single Shot Multi-Box Detector (SSD) with MobileNetV1 and a Faster Region-based Convolutional Neural Network (Faster-RCNN) with InceptionV2. Result shows that one model is ideal for real-time application because of speed and the other can be used for more accurate object detection.

2.1.2 Real-Time Detection and Motion Recognition of Human Moving Objects

Meimei Gong and Yiming Shu have written about Real-Time Detection and Motion Recognition of Human Moving Objects Based on Deep Learning and Multi-Scale Feature Fusion in Video in the year 2020. In their paper, they have proposed a real-time detection positioning and recognition network based on multi-scale feature fusion (IMFF-SSD), which improves the positioning accuracy and identification accuracy. First, this article analyzes the multi-scale features extracted from the SSD network. It combines the position-sensitive information provided by low-level detail features with the context information provided by high-level semantic features through feature fusion, which effectively improves positioning accuracy of the target prediction layer in the SSD network. Secondly, a feature embedded prediction structure is designed to strengthen the semantics of target features without changing the spatial resolution of the SSD prediction layer, and embed low-scale detailed features in high-semantic features for collaborative prediction of targets. This improves the accuracy of the SSD network's recognition of human moving targets at all scales.

2.1.3 You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi have presented us about YOLO in the year 2015. We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, they frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Their unified architecture is extremely fast.

2.1.4 MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun

Wang, Tobias Weyand, Marco Andreetto and Hartwig Adam have put forward about

MobileNets in the year 2017 which is an efficient models for mobile devices and embedded vision applications. These MobileNets are rationalized architecture and it is applied using depth-wise convolutions. Two straight forward global hyper-parameters were introduced by them which are used to efficiently trade between accuracy and latency, so that it grants the model builder the rights to choose the appropriate model for the chosen applications on the basis of the problem's constraints. They have also explained about the potential of MobileNets and effectiveness of MobileNets over various applications. The uses cases of MobileNets includes geo localization, fine grain classification and object detection.

2.1.5 Real-time GPU-based Face Detection in HD Video Sequences

David Oro, Carles Fernandez, Javier Rodriguez Saeta, Xavier Martorell and Javier Hernando in the year 2011 presented us a face detector based on Haar that is optimized and which is applied on high definition videos. The operations by kernel mentioned in their paper makes use of both fine grain parallelism and coarse for computation of integral image and evaluation of filter, therefore it is not only helpful for face detection but it is also beneficial for various computer vision implementations.

2.1.6 A cascade framework for masked face detection

Wei Bu, Jiangjian Xiao, Chuanhong Zhou, Minmin Yang and Chengbin Peng has come up with a fresh CNN-based cascade framework in the year 2018, that comprises of three thoroughly designed CNN to recognize faces that has a mask on it as there is only less face mask data set that detects accurately, due to which the difficulty to detect face mask is increased. In this paper, they have proposed a fresh dataset named "MASKED FACE dataset" to refine the CNN models and examine the face mask detection algorithm developed on the testing set, and obtain desired performance.

2.1.7 Performance Evaluation of Intelligent Face Mask Detection System with various Deep Learning Classifiers

C.Jagadeeswari and M. Uday Theja in year 2020 have highlighted the usage

of deep learning technique to track each individual who fail to wear face mask in their paper. The mentioned system has been instructed to detect precisely the presence of face mask on a person's face. Once the model finds a person without face mask, people around are alerted with an alarm also the authorities responsible are also alerted, by which appropriate actions can be implied against people who are violating the rules. Their paper have also shown a detailed comparison between the classifiers like VGG16, RESNET50, MobileNetV2, each with optimizer SGD, ADAGRAD and ADAM .

2.1.8 A Review on Face Mask Detection using Convolutional Neural Network

Adithya K and Jismi Babu in year 2020 have written a paper about the system they developed. It was an efficient face mask monitoring system using machine learning and image processing analysis. By the development of machine learning and image processing analysis introduce methods for mask detection. By using image processing analysis and machine learning method is used for find out mask detection. Face mask detection can be done through various methods. Mainly convolutional neural network method is used rapidly. The accuracy and decision making is very high in CNN compared to others.

2.1.9 The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China

Kiesha Prem et al. have surveyed and published in the year 2020 on how the mix of population change have affected outbreak in Wuhan. People there have used private location - specified contact patterns and used those ideas in schools, workplace, and reduction of population mixing in general communities. The epidemiological parameters captured and the matrices mentioned above, the latest figures of the Wuhan outbreak is estimated. It has been replicated on the ongoing situations in Wuhan adapting the age-structured susceptible-exposed-infected-removed (SEIR) model for various physical distance measures. They also worked on simulation models to check the dangers and effects of current scenario's outbreak, on how lifting the control measures would affect the people and guiding by allowing

them to return to work in specified stages.

2.1.10 Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques

Narinder Singh Punj, Sanjay Kumar Sondhadra and Sonali Agarwal in their paper in year 2020 have put forward an automated framework built out of deep learning for monitoring social distance between the individuals by using the surveillance camera videos. The individuals are segregated from the background using YOLO v3 object detection model. The identified individual are tracked with bounding boxes and labels assigned using deep sort approach. An experimental analysis was carried out, from which it was noted that the YOLO v3 framework with deep sort tracking approach displayed desired results with balanced mAP and FPS score to detect social distancing violation in real-time.

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

Shashi Yadav in year 2020 has proposed an computer vision based, live automated monitoring system that detects people's both social distancing and face masks in public area by implementing it on raspberry pi4. After any breach detection, the system sends alert to state police headquarters and also alarms the public. In his system deep learning has been mixed with geometric method for developing a robust model. The work here detects both with a accuracy of 85% in best environments.

2.1.12 An Automatic System to Monitor the Physical Distance and Face Mask Wearing of Construction Workers in COVID-19 Pandemic

Moein Razavi, Hamed Alikhani, Vahid Janfaza, Benyamin Sadegh and Ehsan Alikhani in the year 2021 have presented a paper on the system they developed to automatically detect the violation of face mask wearing and physical distancing among construction workers to assure their safety on infrastructure projects during the pandemic. The final model was applied on four videos of road maintenance projects in Houston, TX, that effectively detected the face mask and physical distance. We recommend that construction owners use the proposed system to enhance construction workers' safety in the pandemic situation.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 GOAL OF THE PROJECT

The goal is to train a custom deep learning model to detect whether a person is wearing a mask or not and whether a person is maintaining social distancing. To build a tool that can potentially detect where each person is, and return a bounding box that turns red if the distance between two people is dangerously close and label mask presence. This can be used by governments to analyze the movement of people and alert them if the situation turns serious.

3.2 PROBLEM STATEMENT

3.2.1 Stimulate people on wearing face mask

The challenge is to train object detection models capable of identifying the location of masked faces in an image as well as the location of unmasked faces in the image. Because of which, we can keep a track on people not wearing mask and warn them to wear mask to reduce the chances of spreading infection. These detectors should be robust to noise and provide as little room as possible to accommodate false positives for masks due to the potentially dire consequences that they would lead to. Ideally, they should be fast enough to work well for real-world applications, something we hope to focus on in future rounds of the competition.

3.2.2 Emphasize the importance of social distancing

The challenge here is to train models capable of identifying the distance between people in an image as well as alert when people violate. This helps people in companies and people working in crowded areas to watch and alert people when violating. It makes people understand the importance of following social distancing. Ideally, they should be fast enough to work well for real-world applications, something we hope to focus on in future rounds of the competition.

3.2.3 Coherent system to help

Rather than installing two different detection system to detect mask and social

distancing , here I have brought both in one. It makes it easier for people who wanted to watch their customers and workers for their safety.

3.2.4 Applies to past, present and future

As the proposed system takes input from live camera or either from existing video or image, it can be used in all the cases. The detection system can be applied in live video streaming and note the people who are violating or it can be applied on recorded video or image and detect people who are violating and use these clips or images as evidence to warn the person. The evidences can also be used by police to fine minimum charges to make people obey the rules.

3.3 SCOPE OF THE PROJECT

3.3.1 Existing system

Currently there are many detection systems to detect face mask and for detecting social distancing violation separately and precisely. People managing a large crowd in markets and companies install these systems to prevent violations and to prevent spreading of infections.

Disadvantage of Existing System:

- Purchasing of two systems
- More memory consumed
- Difficult to maintain both separately

3.3.2 Proposed system

In this proposed system, the face mask detection and social distancing detection is done in single slate. People managing a large crowd in markets and companies install this one system to detect both for reducing the violations and to prevent spreading of infections.

Advantages of Proposed System:

- Purchasing just one system for both detection

Less memory consumption

Easy to maintain and keep a track

3.3.2.1 General flow of the proposed system

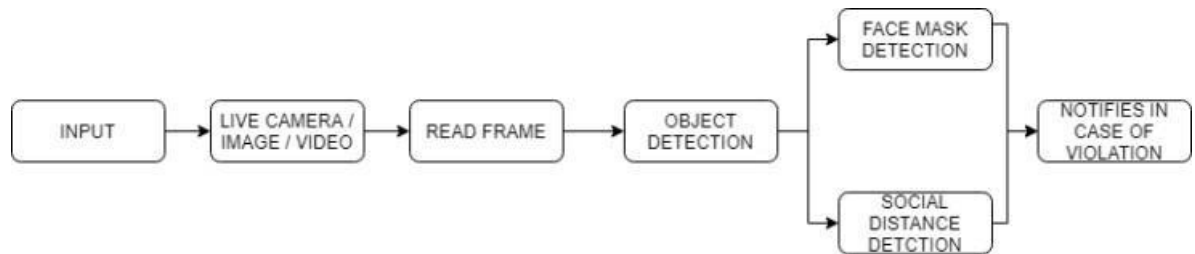


Fig 3.1: General flow of the proposed system

The flow diagram above show the general worki ng of the proposed system. The system firsts get input from live camera or either in image or video formats. It then reads frame from the input. The object is detected from the frames. If there is a presence of an object, it checks for the face mask and social distance is calculated between each object. If violation occurs, it notifies about the violation.

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

4.1 SYSTEM DEVELOPEMENT ENVIRONMENT

4.1.1 Introduction to python

PYTHON is an interpreted, high level, general purpose programming language. we can use python for developing desktop GUI application, websites and web application by taki ng care of common programming tasks. Python is the programming language used in the code examples throughout this book. Python is a clear and concise language with good support for input/output, numeric, images and plotting. The language has some peculiarities such as indentation and compact syntax that takes getting used to.

Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

4.1.2 Introduction to computer vision

Computer Vision, often abbreviated as CV, is defined as a field of study that seeks to develop techniques to help computers “see” and understand the content of digital images such as photographs and videos.

If we parallelize the human and computer vision systems, we can say that both consist of a sensor and an interpreter. When it comes to sensors, cameras are most often considered the equivalent of the eyes for a computer vision system. In reality, cameras are only a part of a wide range of hardware that are used to capture information about the real world.

It is a multidisciplinary field that could broadly be called a subfield of artificial intelligence and machine learning, which may involve the use of specialized methods

and make use of general learning algorithms.

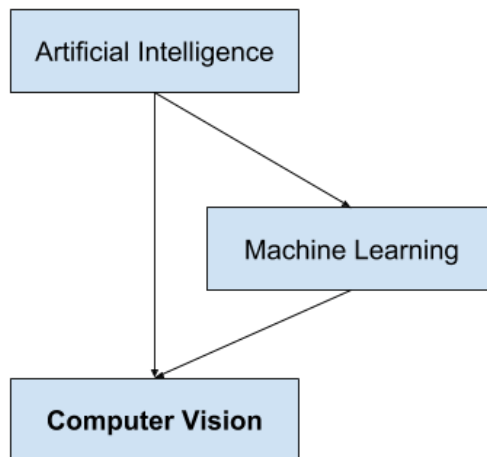


Fig 4.1: Computer vision Conceptual diagram

4.1.3 Introduction to deep learning

Though the technology is often considered a set of algorithms which ‘mimics the brain’, a more appropriate description would be a set of algorithms which ‘learns in layers’. It involves learning through layers that enable a computer to develop a hierarchy of complicated concepts from simpler concepts.

To understand what deep learning is, we first need to understand the relationship deep learning has with machine learning, neural networks, and artificial intelligence.

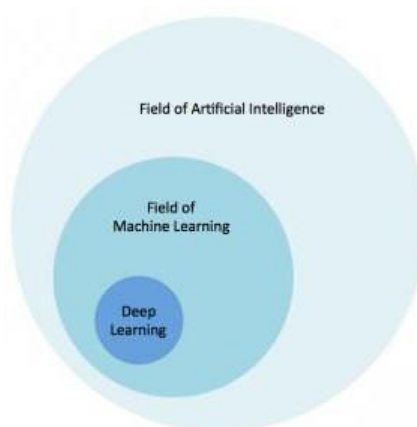


Fig 4.2: Deep Learning Relationship diagram

4.1.4 Introduction to image classification

Image classification is a supervised learning problem: define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos. Early computer vision models relied on raw pixel data as the input to the model. However, as shown in Figure , raw pixel data alone doesn't provide a sufficiently stable representation to encompass the myriad variations of an object as captured in an image. The position of the object, background behind the object, ambient lighting, camera angle, and camera focus all can produce fluctuation in raw pixel data; these differences are significant enough that they cannot be corrected for by taking weighted averages of pixel RGB values.

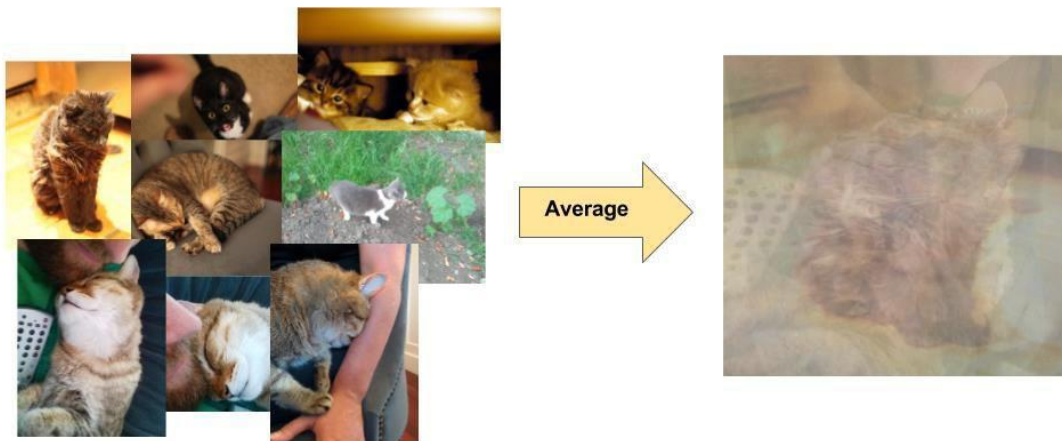


Fig 4.3: Image classification example

To model objects more flexibly, classic computer vision models added new features derived from pixel data, such as color histograms, textures, and shapes.

4.1.4.1 General Approaches to Image Classification

1. Unsupervised

Unsupervised classification (a.k.a., “clustering”) identifies groups of pixels that exhibit a similar spectral response

2. Supervised

Supervised classification uses image pixels representing regions of known, homogeneous surface composition -- training areas -- to classify unknown pixels.

4.1.5 Introduction to object detection

Object detection is a computer vision technique that allows us to identify and locate objects in an image or video. With this kind of identification and localization, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labeling them

Object detection is inextricably linked to other similar computer vision techniques like image recognition and image segmentation, in that it helps us understand and analyze scenes in images or video.

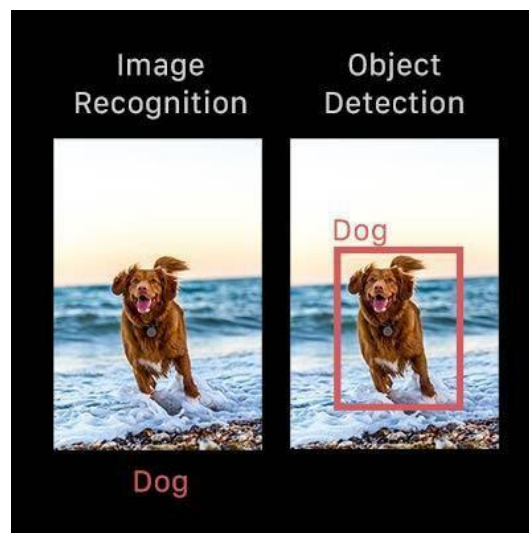


Fig 4.4: Object detection example

Broadly speaking, object detection can be broken down into machine learning-based approaches and deep learning-based approaches.

Deep learning-based approaches employ convolutional neural networks (CNNs) to perform end-to-end, unsupervised object detection, in which features don't

need to be defined and extracted separately.

4.2 SYSTEM REQUIREMENTS

4.2.1 Hardware requirements

Table 4.1: Hardware requirements

1.	Processor	:Intel Core i5-5300U CPU @2.30GHz
2.	Operating System	:64-bit OS
3.	RAM	:Minimum of 8.00 GB
4.	GPU	:Intel HD Graphics 5500
5.	Camera	:Laptop's in built camera or external web cam

4.2.2 Software requirements

Table 4.2: Software requirements

1.	Python	3.6.10
2.	OpenCV	4.2.0
3.	Tensorflow	1.15.0
4.	Keras	2.3.1
5.	numpy	1.19.1

4.3 SOFTWARE REQUIREMENT SPECIFICATION

The purpose of this Software Requirement Specification (SRS) is to help the project.

It is provided with some requirements which are used in the Transaction Mercator System. All parts design, coding and testing will be prepared with helping of SRS.

4.3.1 Library software

The project is built using Python as a platform and OpenCV as a main source. We have also used certain libraries.

1. **OPENCV(Open Source Computer Vision Library)** is a python wrapper for the original opencv c++ implementation. The library is cross platform and free for use under the opensource BSD license. Opencv is capable of image analysis and processing but have more like capture from cam. This means it is great at taking frames out of video or taking in two frames from a stereoscopic camera and run algorithm to extract information
2. **Keras** is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

However, Keras is also a highly-flexible framework suitable to iterate on state-of-the-art research ideas. Keras follows the principle of progressive disclosure of complexity: it makes it easy to get started, yet it makes it possible to handle arbitrarily advanced use cases, only requiring incremental learning at each step.

In much the same way that you were able to train & evaluate a simple neural network above in a few lines, you can use Keras to quickly develop new training procedures or exotic model architectures.

3. **TensorFlow** is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications.

4. **NumPy** is one of the core libraries in Python programming and provides support for arrays. An image is essentially a standard Numpy array containing pixels of data points. Therefore, by using basic NumPy operations, such as slicing, masking and fancy indexing, we can modify the pixel values of an image. The image can be loaded using `skimage` and displayed using `matplotlib`.

Furthermore, NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better known under the heading of "big data". Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

4.3.1.1 Methods Included

Few major Methods that has been used in our project has been mentioned below. The methods are essential to construct a stable system.

VIDEOCAPTURE() this method grabs the next frame from video file or camera and the return true(non-zero) in the case of success..that is,we call `videocapture ::grab()` for each camera and after that call the `videocapture::retrieve()` to decode and get frame from each camera.

IMREAD() used to avoid data structure and function name conflicts with others libraries,**opencv** has its own namespace: `cv`..Then create a `mat` object that will store the data of the loaded image.`mat image`;now we call the **imread** function which loads the image name specified by the first argument(`argv[1]`).use the function **cv2.imread()** to read an image.the image should be in the working directory or a full path or image should be given.

IMSHOW(I) displays the grayscale image `I` in a figure.**imshow** uses the default display range for the image data type and optimize figure,axes and image object properties for image display..**imshow(RGB)** displays the truecolor image `RGB` in a figure. **imshow(BW)** displays the binary image `BW` in a figure.use the function

cv2.imshow() to display an image in a window. The window automatically fits to the

image size.

WAITKEY() is a keyboard binding function in opencv it is used to introduce a delay of n milliseconds while rendering images to windows. Its argument is the time in milliseconds for any keyboard event. If you press any key in that time, the program continues. If 0 is passed, it waits indefinitely for a key stroke. It can also be set to detect specific key strokes like, if key 'a' is pressed etc.

DESTROYALLWINDOWS() simply destroys all the windows we created. If you want to destroy any specific windows, use the function `cv2.destroyAllWindows()` where you pass the exact window name as the argument.

4.4 MODULE IMPLEMENTATION

The detection system comprises of two modules. The Face mask detection system is one and Social distancing detection system is another. The work flow of the modules mentioned above is briefed below.

4.4.1 Face mask detection system

Face mask detection system is a separate module which detects the presence of face mask on people's face. The module is broken down into two phases.

1. **Training:** Here the mask detection data set is loaded from the disk, model is trained on the data set and the mask detector is serialized to the disk.
2. **Deployment:** Once it is trained, the mask detector is loaded, face detection is performed and it is classified as mask or no mask.

A face mask dataset is prepared and trained for image classification. The trained classifier is then loaded for detection. To detect face mask, we get input from the user in various formats. The frames are read from the video input and

the faces are detected from the input. The Region Of Interest (ROI) of each face

is extracted and the mask classifier is implemented to each ROI. The presence of face mask is checked and the results are shown. It labels “Mask” if there is a mask on a person’s face and if not, it labels “No Mask”.

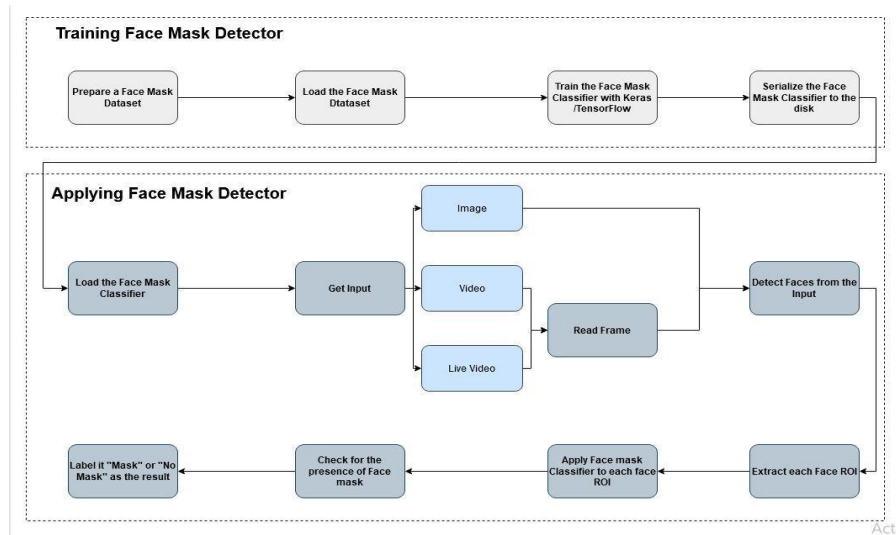


Fig 4.5: Face Mask detection Flow diagram

4.4.1.1 Caffe model - Face detection

The system needs to detect the faces initially to check for the presence of mask. Therefore, I have implemented face detection using Caffe model. The Caffe model is based on the SSD and it uses ResNet architecture as its backbone. This model can process over 60 million image in single day which makes it easier to train and detect with greater accuracy.

Caffe is a deep learning framework made with expression, speed, and modularity in mind. Deep networks are compositional models that are naturally represented as a collection of inter-connected layers that work on chunks of data. Caffe defines a net layer-by-layer in its own model schema. The network defines the entire model bottom-to-top from input data to loss. As data and derivatives flow through the network in the forward and backward passes Caffe stores, communicates, and manipulates the information as *blobs*: the blob is the standard

array and unified memory interface for the framework. The layer comes next as the

foundation of both model and computation. The net follows as the collection and connection of layers. The details of blob describe how information is stored and communicated in and across layers and nets.

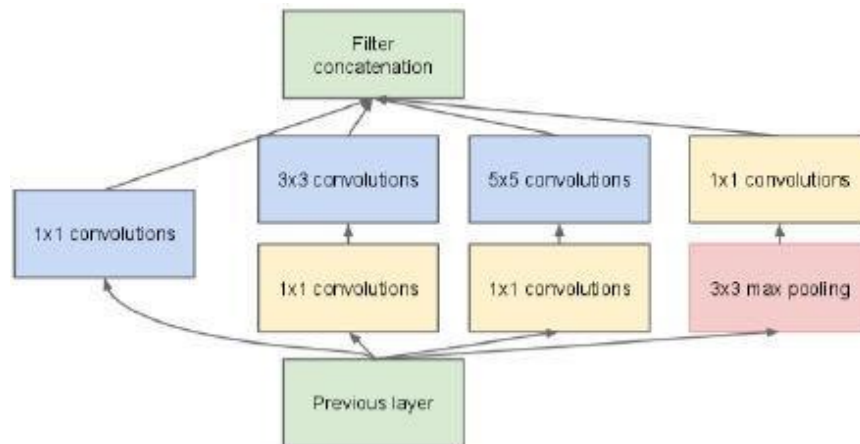


Fig 4.6: Caffe Model Architecture

Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU. That's 1 ms/image for inference and 4 ms/image for learning and more recent library versions and hardware are faster still. We believe that Caffe is among the fastest convnet implementations available.

4.4.1.2 MobileNetV2 - Human classification

Once the face detection is done precisely, the face mask detection has to be done. To detect the face mask, we need to apply image classification to classify faces with mask and without mask. This is performed with MobileNetV2. It is a deep CNN with 53 layers. We can use “classify” to classify new set of images using MobileNetV2

model.

MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. We shall be using Mobilenet as it is lightweight in its architecture. It uses depth wise separable convolutions which basically means it performs a single convolution on each colour channel rather than combining all three and flattening it. We then demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, fine grain classification, face attributes and large scale geo-localization.

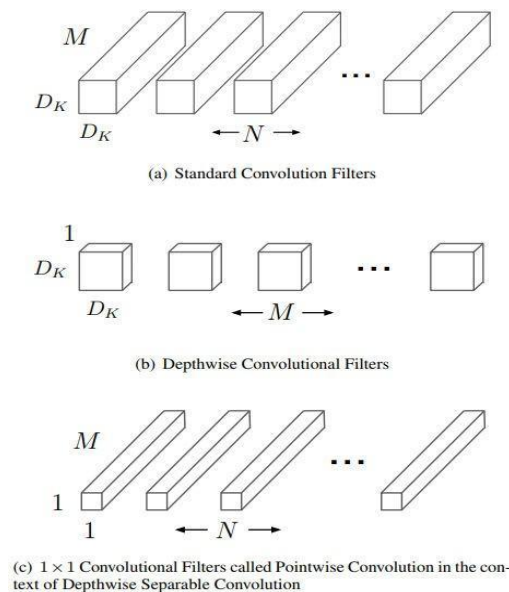


Fig 4.7: Point wise and depth wise convolutions



```
03 - val_loss: 0.2933 - val_acc: 0.9205
Epoch 20/20
94/95 [=====] - ETA: 1s - loss: 0.0787 - acc: 0.9740Epoch 1/20
767/95 [=====]
=====
=====] - 20s 27ms/sample - loss: 0.3165 - acc: 0.9048
95/95 [=====] - 134s 1s/step - loss: 0.0784 - acc: 0.9740 - val_loss: 0.3556 - val_acc: 0.9048
[INFO] evaluating network...
      precision    recall  f1-score   support

with_mask         1.00      0.81      0.89       383
without_mask      0.84      1.00      0.91       384

   accuracy
macro avg      0.92      0.90      0.90       767
weighted avg   0.92      0.90      0.90       767

[INFO] saving mask detector model...
Oobject::moveToThread: Current thread (0x40b2700) is not the object's thread (0x
```

Fig 4.8: Face mask detection model accuracy

It shows the accuracy and loss of the face mask detection model. The generated model's accuracy is calculated and it is around 90%.

4.4.2 Social distancing detection system

Social distance detection system is another module which detects the distance between the people and notify if there is a violation.

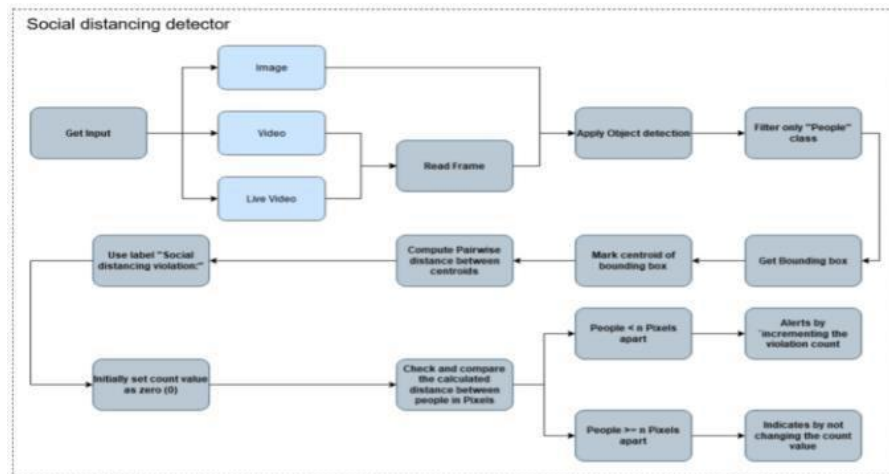


Fig 4.9: Social distance detection Flow diagram

The social distance detection system gets input from the surveillance camera or from user either in image or video format. The object detection is applied and "People" class is filtered.

The model generates a set of bounding box around each human. Each individual is correlated with 3-dimensional feature area (x,y,d), where (x,y) refers to the centroid points of the displayed bounding box and d refers to the individual's depth that we see from the surveillance camera or from the video or image that we give as input.

$$d = ((2 * 3.14 * 180) / (w + h * 360)) * 1000 + 3$$
, where the bounding box's width is referred as w and its height is referred as h.

Pairwise L2 norm is estimated for the group of bounding boxes..The L2 norm's dense matrix is applied to mark the contiguous one for each person that makes certain of the closeness. If any person that satisfies closeness property is marked as a neighbour. The contiguous ones are represented in discrete color and the social distancing violation count is incremented. If there is no violation, the count value remains null.

4.4.2.1 YOLO- Object detection

In case of social distance detection, we need to detect the object first. In our case, persons are the object. We have used YOLO for object detection. It processes images in real time at minimum of 45 frames per second and maximum of around 155 frames per second. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection.

First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. Second, YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin.

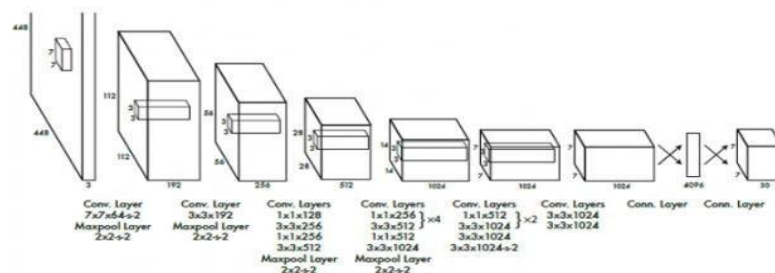


Fig 4.10: You Only Look Once architecture

4.5 FLOW OF THE PROPOSED SYSTEM

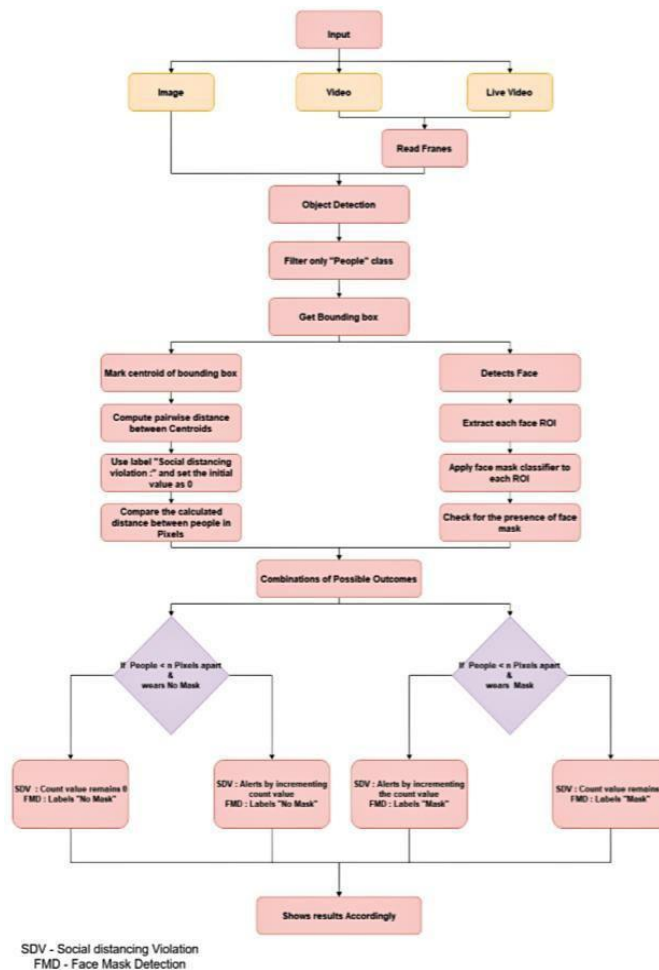


Fig 4.11: Flow of the proposed system

The modules involved in this detection system is discussed previously. Those modules are brought together and made to work as single system. The overall proposed system detects both face mask and social distance violation detection at single point of control.

The proposed system makes it easier to detect the violation and it consumes less time by showing the output of each module simultaneously rather than implementing each module separately.

CHAPTER 5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 INTRODUCTION

Facial recognition and object detection systems use computer vision technology to identify individuals for security purposes. Most of the practices have positive changes in some metrics after adopting the face mask and social distance detection system. The current scenario made people realize the need of the system to follow precautionary methods.

5.2 REQUIREMENT ANALYSIS

The developer is responsible for:

Developing the system, which meets the SRS and solving all the requirements of the system?

- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

5.2.1 Functional requirements

Functional requirements define what a software system should do. It defines a function of a software system or its module. Functionality is measured as a set of inputs to the system under test to the output from the system.

Functional requirements could include the following components that can be measured as part of functional testing:

- 1) **Interoperability:** Requirement describes whether a software system is interoperable across different systems.

- 2) **Security:** The functional requirement describes the security aspect of software requirements.
- 3) **Accuracy:** Accuracy defines a data entered into the system is correctly calculated and used by the system and that the output is correct.
- 4) **Compliance:** Compliance functional requirements validate that the developed system is compliant to Industrial standards.

5.2.2 Non-functional requirements

The non-functional requirement says about “what a system should be” rather than “what a system should do” (functional requirement). They are mostly derived from functional requirements based on input from the customer and other stakeholders. Non-functional requirements consist of below sub types (non-exhaustive):

- 1) **Performance:** A performance attribute type of non-functional requirement measures system performance. Please remember that system performance measurement is different from load measurement. During load testing, we load the system CPU and RAM and check the system throughput. In the case of performance, we test system throughput in normal load/stress conditions.
- 2) **Usability:** Usability measures the usability of the software system being developed.
- 3) **Maintainability:** Maintainability of a software system is the ease with which the system can be maintained. If the Mean Time Between Failures (MTBF) is low or Mean Time To Repair (MTTR) is high for the system being developed, then the maintainability of the system is considered low. Maintainability is often measured at code level using Cyclomatic complexity. Cyclomatic complexity says that the lesser complex the code is, the easier it is to maintain the software.
- 4) **Reliability:** Reliability is another aspect of availability. This quality attribute emphasizes the availability of a system under certain conditions. It is measured as MTBF just like maintainability.
- 5) **Portability:** Portability means the ability of a software system to work in a

different environment if the underlying dependent framework stays the same.

6) **Serviceability:**Serviceability of a software system is the ability of a service/technical expert to install the software system in a real-time environment, monitor the system while it is running, identify any technical issues in the system and provide a solution to resolve the issue.Serviceability is possible if the system is developed to facilitate serviceability.

7) **Adaptability:**The adaptability of a system is defined as the ability of a software system to adapt to change in an environment without any change in its behavior.

5.2.3 Performance requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the required specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the required specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

5.3 SYSTEM ANALYSIS

5.3.1 Introduction to system analysis

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and another is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then the successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult

and requires creative thinking and understanding of existing running system is also difficult, improper understanding of the present system can lead diversion

from solution.

5.3.2 Analysis model

The model that is basically being followed is the SPIRAL MODEL, which states that the phases are organized in a linear order. First of all, the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of the new module is needed, analysis of the present system can be used as the basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project is Requirement Analysis, Project Planning, System design, Detail design, Coding, Unit System integration & testing and here the linear ordering of these activities is critical. End of the phase and the output of testing .

5.3.2.1 Spiral model

One phase is the input of another phase. It was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

The spiral model is a combination of sequential and prototype models. This model is best used for large projects which involve continuous enhancements. There are specific activities that are done in one iteration (spiral) where the output is a small prototype of the large software. The same activities are then repeated for all the spirals until the entire software is built.

This model is very good to use for larger projects where you can develop

and deliver smaller prototypes and can enhance it to make the larger software. The implementation of this model requires experienced resources as risk analysis is a very integral part of this model and risk analysis requires expertise and as a result this model becomes costly.

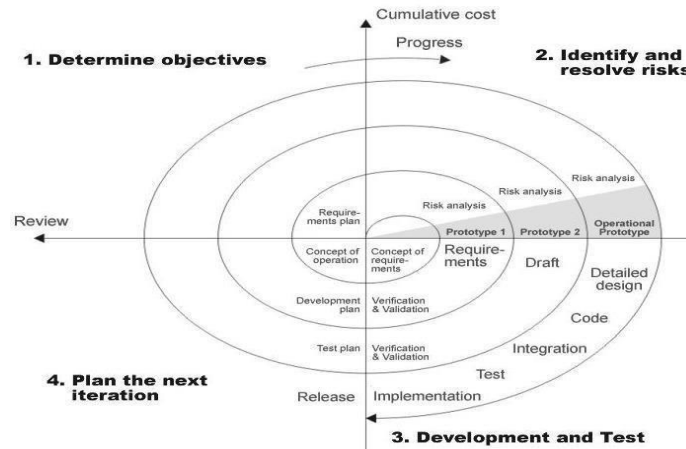


Fig 5.1: Analysis Model

5.4 SYSTEM DESIGN

The proposed system is an efficient detection system which detects the presence of face mask on each person's face and whether there is social distancing violations. It detects from the live video surveillance, existing images and videos which makes it unique from other existing systems. The system proposed here detects both simultaneously and far more accurately. It ensures public's safety.

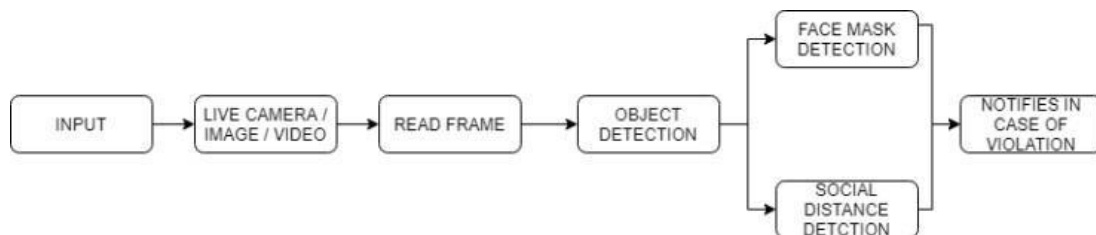


Fig 5.2: General flow of the system

The above figure is the generalized flow of the project. The system firsts get input from live camera or either in image or video formats. It then reads frame from the input. The object is detected from the frames. If there is a presence of an object, it checks for the face mask and social distance is calculated between each object. If violation occurs, it notifies about the violation.

5.5 SYSTEM EVOLUTION

The detection system I have proposed will recognize people wearing face mask and maintaining social distancing in a single slate. It detects from live video and also from existing video or images. It also alerts and notifies when someone violates the rules.

5.5.1 System to be changed:

There may be a chance of missing few faces to detect in a highly crowded environment. The accuracy of the project maybe me of 85% to 95% on overall. The accuracy can be up to 99% in better environment and better angle of camera coverage. The system's accuracy can be improved further by training the the model with larger dataset.

5.5.2 Change Proposal:

The change proposal to the proposed system is to develop it into a full stack application, that can get the details of people who are violating. Body gesture can be included to this system to detect coughing and sneezing. As fever is also another threat, the system can be equipped to any thermal screening cameras to detect the body temperature.

5.5.3 System understanding:

Complete understanding of the system that is to be generated i.e. a brief study of the requirements and Designing the system that is to be developed.

5.5.4 System Validation:

Validation can be fined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer, i.e. the customer expected output i.e. fulfilling all the customer specified requirements.

In our case, the customers are people handling huge crowd, like company

managers, shop owners, stations,..etc.

5.5.5 Modified System:

Modified system provides a accurate detection of both mask and social distancing. It can be made available for everyone by developing it into an application.

5.6 RESULT

As the proposed system can be applied on three different formats of input, the outputs of each kind of input is disclosed below.

5.6.1 Image as Input

The proposed system output for the image input initially process the input image and detects the individual and their face from it. Then the people and their face is confined in the bounding boxes with the confidence score. The colour of the bounding box is changed to red when there is a violation and the count value is increased when there is a social distancing violation.

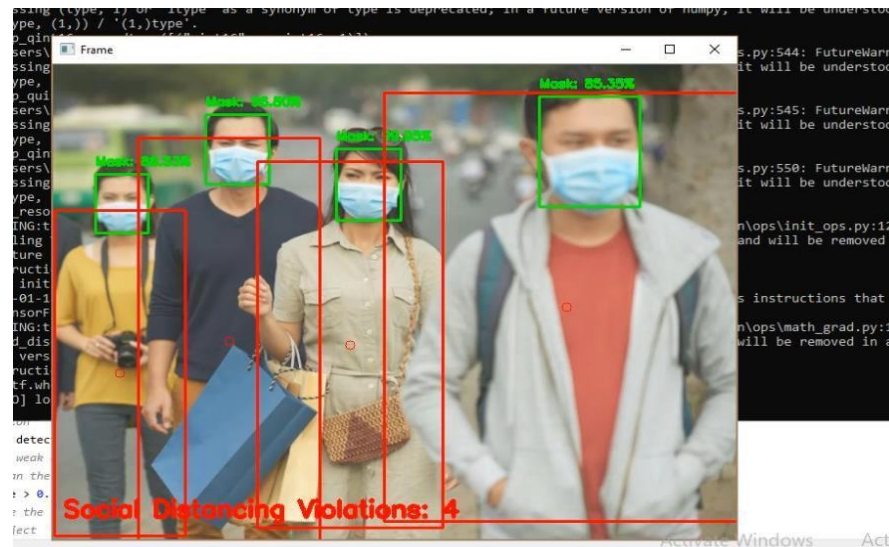


Fig 5.3: People wearing face mask and violating social distance in an Image

In the Fig 5.3, it shows that the people are violating social distancing but all are wearing face mask.

5.6.2 Video as Input

The outputs for the video input reads the frames from the video and detects

the person and face of the person from it. Then the person's face is confined in the bounding boxes with the confidence score. The bounding boxes remain in green colour if the people are following the measures. The bounding boxes change colour to red when there is a violation and the count value is incremented when there is a social distancing violation.

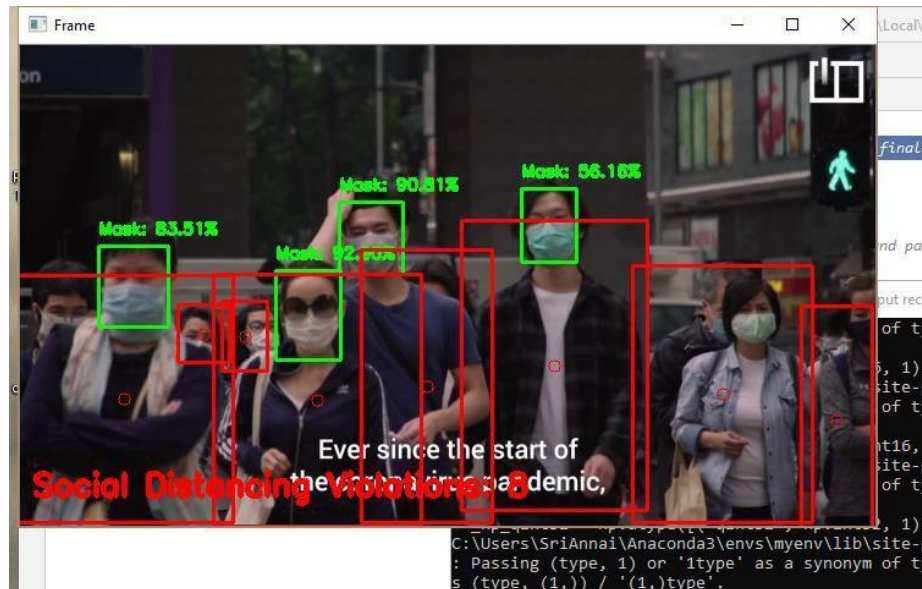


Fig 5.4: Detection of violation in a video input

In the Fig 5.4, it shows the screen shot of the video on which the system has detected the violations. We can see that, there are few people who are not wearing a mask and many are not following social distancing. It also shows the people who are wearing a mask with green boxes around their face.

5.6.3 Input from Live camera

The outputs for the live camera input reads the frames from the streaming video and detects the person and face of the person from it. Then the person's face is confined in the bounding boxes with the confidence score. The bounding boxes remain in green colour if the people are following the measures. The bounding boxes change colour to red when there is a violation and the count value is incremented when there is a social distancing violation.

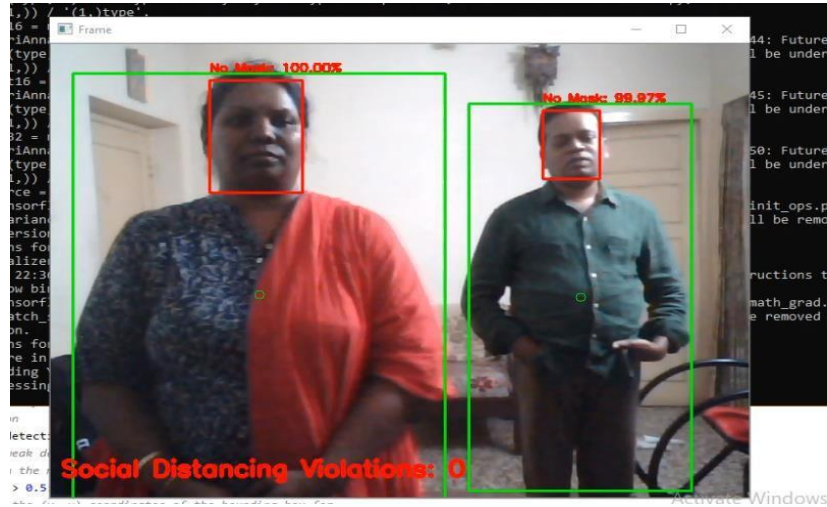


Fig 5.5: Violation detection on a live camera

In the Fig 5.6, it shows the screen shot of the live video on which the system has detected the violations. We can see that, both of them is not wearing a mask and it also detects that there is no social distancing violation.

CHAPTER 6

SUMMARY

AND

CONCLUSION

6.1 INTRODUCTION

This Chapter describes and discuss the objectives of the system stipulated in earlier chapters, limitation of the system conclusion and recommendation of the proposed system.

6.2 SUMMARY

As discussed in the previous chapters the main problem that we addressed was accuracy and efficiency of both the detection. Therefore, it motivated me to develop both precautionary detection in a single state and enable them to handle work efficiently and effectively. The Project has implemented most of the objectives stipulated in earlier chapter. The Face mask and social distancing detection system offers a number of benefits to the user and it facilitates to prevent others and themselves from infections.

6.3 CONCLUSION

I have put forward a computer vision based system which ensures people's safety in public places from COVID-19 virus infection. It is done by assisting firms, any organizations or government to surveillance through camera equipped with the proposed system. The system can also be applied on any images or videos.

The current study used OpenCV,Python and CNN to detect whether people were wearing face masks or not. The models were tested with images and real-time video streams. The accuracy of the model on average is around 95%, the optimization of the model is a continuous process and we are building a highly accurate solution by tuning the hyper parameters.

An efficient real-time deep learning based framework to automate the process of monitoring the social distancing via object detection and tracking approaches, where each individual is identified in the real-time with the help of bounding boxes.

The generated bounding boxes aid in identifying the clusters or groups of people satisfying the closeness property computed with the help of pairwise vectorized approach. The accuracy of the social distancing alone will be nearly 90%.

Thus, the system function in an efficient way in the current scenario where the virus is spreading faster and the vaccine has not yet suggested as the only sufficient way to prevent ourselves from the deadly virus. Therefore, detection of social distancing and presence of face mask violations helps in ensuring human health's safety. The solution I have proposed here saves time and reduces the spread of the coronavirus. The accuracy of the system is around 85% to 95% and it can reach around 99% based on the surrounding environment. This system can be applied on places with huge crowd like malls, stations, companies, streets, etc.

6.4 FUTURE WORK

The system has only some features that will help in attaining a complete solution for the current problem. There are other use cases that can be added to derive a more efficient system for the safety of people.

The system proposed here can be developed into a full stack application, that can get the data of each person violating the precautionary measures. The details can be used to send a warning message to the person or fine a minimum amount to let them know the importance of following the measures.

As coughing and sneezing is the cause for spreading of COVID-19 infection, including body gesture to detect coughing or sneezing by a person in a crowded place will help the current situation.

Another symptom of corona virus is body temperature, it is difficult and vulnerable of getting infected to note the temperature of large crowd of people with IR thermometers. This system can be equipped to any thermal screening cameras to detect the body temperature of each person in public areas effectively.

REFERENCE

1. Adithya K, Jismi Babu, "A Review on Face Mask Detection using Convolutional Neural Network", IJRASET, vol. 7, no. 11, pp. 1302-1304, 2020.
2. Ake Frankenfield, "Artificial Intelligence (AI)", <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp> , Investopedia, 2021.
3. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv, 2017.
4. Anwaar Ulhaq, Jannis Born, Asim Khan, Douglas Pinto Sampaio Gomes, Subrata Chakraborty, Manoranjan Paul, "COVID-19 Control by computer vision approaches: A Survey", IEEE, vol. 8, pp. 2169-3536, 2020.
5. C.Jagadeeswari and M.Uday Theja, "Performance Evaluation of Intelligent Face Mask Detection System with Various Deep Learning Classifiers", International Journal of Advanced Science and Technology (IJAST), vol. 29, no. 11s, pp. 3074-3082, 2020.
6. Chen S., Zhang C., Dong M., Le J., Rao M., "Using ranking-cnn for age estimates", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
7. Chen Wang, Horby Peter W, Hayden Frederick G, Gao George F, "A novel coronavirus outbreak of global health concern", The Lancet, vol. 395, no. 10223, pp. 470-473, 2020.
8. David Oro, Carles Fernández, Javier Rodríguez Saeta, Xavier Martorell and Javier Hernando, "Real-time GPU-based face detection in HD video sequences", IEEE International Conference on Computer Vision Workshops, 2011.
9. Ilija Mihajlovic, "Everything you ever wanted to know about Computer Vision", <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e> , 2019.

10. J. R. A. Farhadi and J. Redmon, "Yolov3: An incremental improvement", arXiv, vol. 17, 2018.

11. Joseph Redmon, Santosh Divval, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
12. K. A. Joshi and D. G. Thakore, "A survey on moving object detection and tracking in video surveillance system", International Journal of Soft Computing and Engineering, vol. 2, no. 3, pp. 44–48, 2012.
13. K. Prem, Y. Liu, T. W. Russell, A. J. Kucharski, R. M. Eggo, N. Davies, S. Flasche, S. Clifford, C. A. Pearson, J. D. Munday et al., "The effect of control strategies to reduce social mixing on outcomes of the covid-19 epidemic in wuhan, china: a modelling study", The Lancet Public Health, 2020.
14. L. Hensley, "Social distancing is out, physical distancing is in, here's how to do it", Global News–Canada (27 March 2020), 2020.
15. Liu H., Chen Z., Li Z., and Hu W., "A powerful method of pedestrian detection based on YOLOv2", Mathematical Problems in Engineering, <https://doi.org/10.1155/2018/3518959>, 2018.
16. Marshall Hargrave, "Deep Learning", <https://www.investopedia.com/terms/d/deep-learning.asp>, Investopedia, 2020.
17. Math Works, "What is Deep Learning?", <https://www.mathworks.com/discovery/deep-learning.html>, [Online; accessed in 2021].
18. Matrajt L, Leung T, "Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease", Emerg Infect Dis, vol. 26(8), pp. 1740-1748, 2020.
19. Narinder Singh Punj, Sanjay Kumar Sondhadra and Sonali Agarwal, "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLOv3 and Deepsort techniques", arXiv, 2020.
20. P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72, 2005.

21. Pias, "Object detection and distance measurement", <https://github.com/paul-pias/Object-Detection-and-Distance-Measurement>, 2020, [Online;accessed 01-March-2020].
22. R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a se crowd", IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2008.
23. Sanchari Sinha Dutta, "Mask-wearing together with social distancing has maximum impact on COVID-19", News Medical (20 Jan 2021), 2021.
24. Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residues and linear bottlenecks", IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, 2018.
25. Shashi Yadav, "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence", IJRASET, vol. 8, no. 7, 2020.
26. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," Springer: European conference on computer vision, pp. 21–37, 2016.
27. W. H. Organization, "WHO corona-viruses (COVID -19)", <https://www.who.int/emergencies/diseases/novel-corona-virus-2019>, 2020, [Online;accessed May 02, 2020].
28. Wei Bu, Jiangjinn Xiao, Chuanhong Zhou, Minmin yang and Chengbin "A Cascade Framework for Masked Face Detection", IEEE 8th International Conference on CTS & RAM, 2017.
29. WHO, "Who director-generals opening remarks at the media briefing on covid-19", <https://www.who.int/dg/speeches/detail/> , 2020, [Online; accessed March 12, 2020].
30. Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review", IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

APPENDICES

A. SOURCE CODE

```
from config import s_d_config as config
from config.detection import detect_people
from scipy.spatial import distance as dist
import numpy as np
import argparse
import imutils
import cv2
import os

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 time

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())
def detect_and_predict_mask(frame, faceNet, maskNet):
    (h, w) = frame.shape[:2]
```

```
blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300),  
                             (104.0, 177.0, 123.0))  
faceNet.setInput(blob)
```

```

detections =
faceNet.forward() faces = []
locs = []
preds = []
for i in range(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]
    if confidence > 0.5:
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")
        (startX, startY) = (max(0, startX), max(0, startY))
        (endX, endY) = (min(w - 1, endX), min(h - 1, endY))
        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)
        faces.append(face)
        locs.append((startX, startY, endX,
endY)) if len(faces) > 0:
            faces = np.array(faces, dtype="float32")
            preds = maskNet.predict(faces, batch_size=32)
    return (locs, preds)

prototxtPath = "face_detector/deploy.prototxt"
weightsPath = "face_detector/res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

maskNet = load_model("mask_det.model")
labelsPath = os.path.sep.join([config.MODEL_PATH, "coco.names"])
LABELS = open(labelsPath).read().strip().split("\n")

```

```
weightsPath = os.path.sep.join([config.MODEL_PATH, "yolov3.weights"])
configPath = os.path.sep.join([config.MODEL_PATH, "yolov3.cfg"])
```

```

print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]
print("[INFO] accessing video stream...")
vs = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output2.avi',fourcc, 20.0, (640,480))
while True:
    (grabbed, frame) = vs.read()
    if grabbed == True:

```

```

        c
        o
    else: p
        y
        =
        f
        r
        a
        m
        e
        .
        c
        o
        p
        y
        (
        )
        o
        u

```

```

t                                     c
.                                     o
w                                     p
r                                     y
i                                     )
t
e                                     break
(

```

```

frame = imutils.resize(frame, width=700)
(locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
for (box, pred) in zip(locs, preds):
    (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred
    label = "Mask" if mask > withoutMask else "No Mask"
    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
    label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
    cv2.putText(frame, label, (startX, startY - 10),
                 cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
    cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
results = detect_people(frame, net, ln,
                        personIdx=LABELS.index("person"))
violate = set()
if len(results) >= 2:

```

```

centroids = np.array([r[2] for r in results])
D = dist.cdist(centroids, centroids, metric="euclidean")
for i in range(0, D.shape[0]):
    for j in range(i + 1, D.shape[1]):

        if D[i, j] < config.MIN_DISTANCE:
            violate.add(i)
            violate.add(j)
for (i, (prob, bbox, centroid)) in enumerate(results):
    (startX, startY, endX, endY) = bbox
    (cX, cY) = centroid
    color = (0, 255, 0)

    if i in violate:
        color = (0, 0, 255)
    cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
    cv2.circle(frame, (cX, cY), 5, color, 1)
    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)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

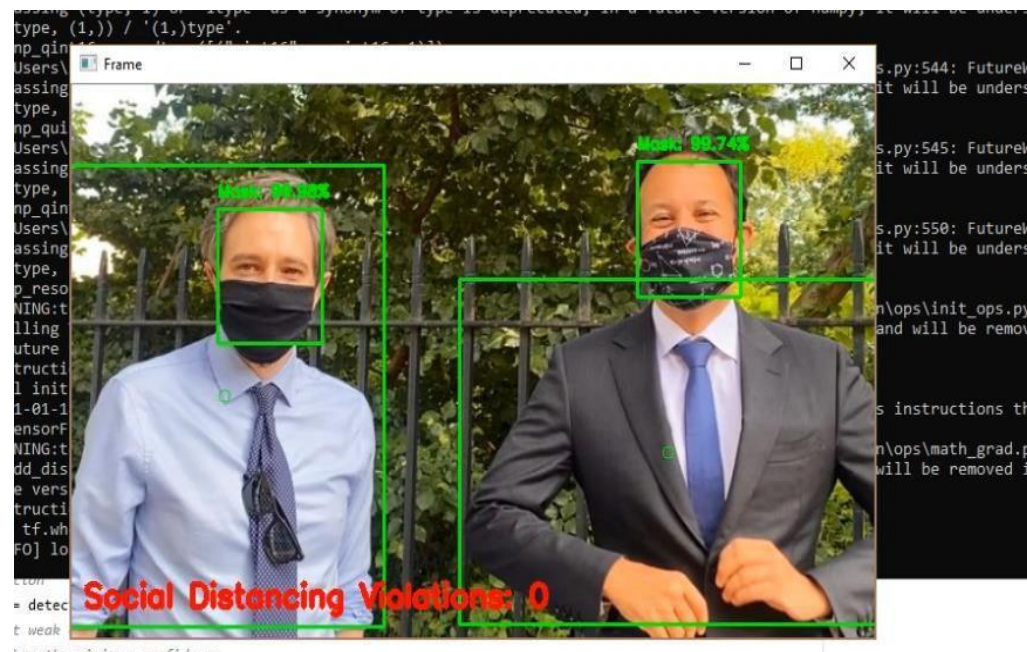
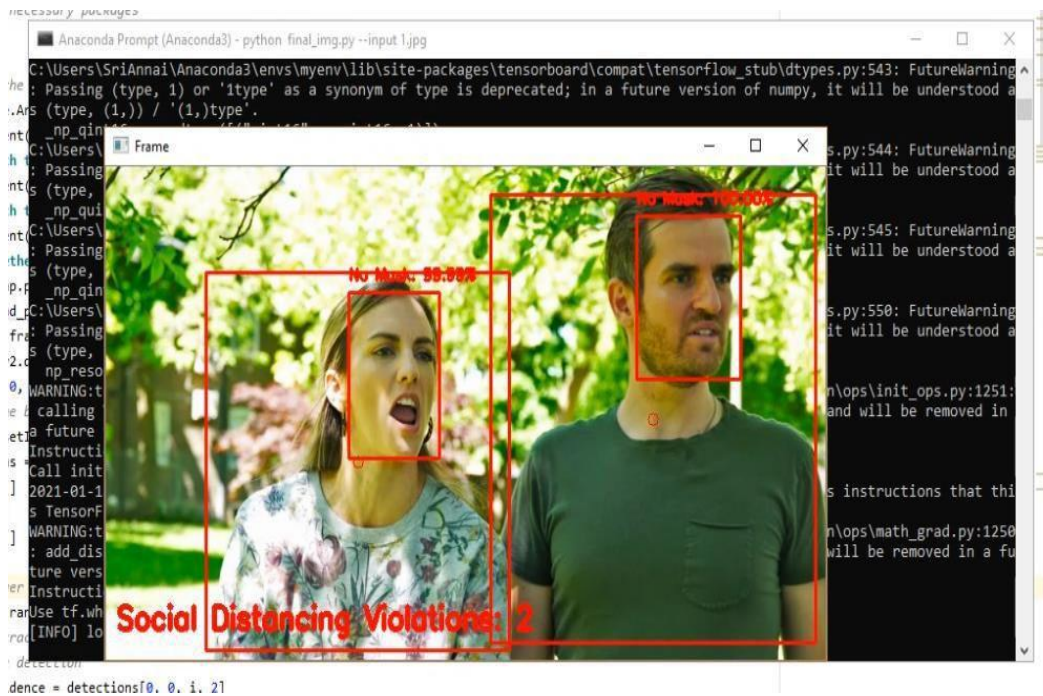
    if key == ord("q"):
        break
vs.release()
out.release()

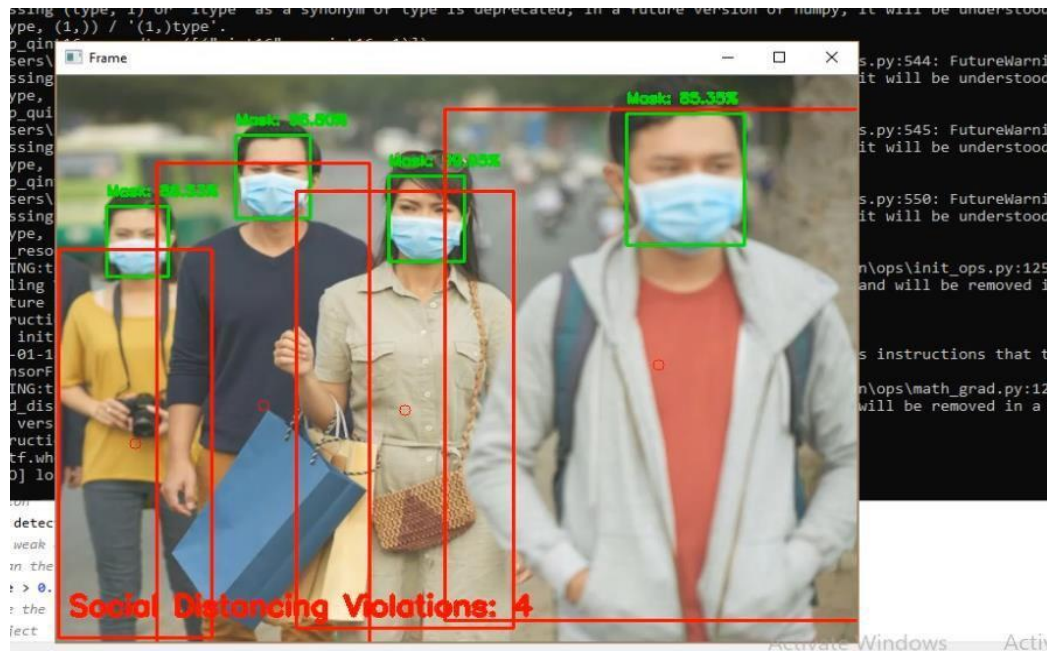
cv2.destroyAllWindows()

```

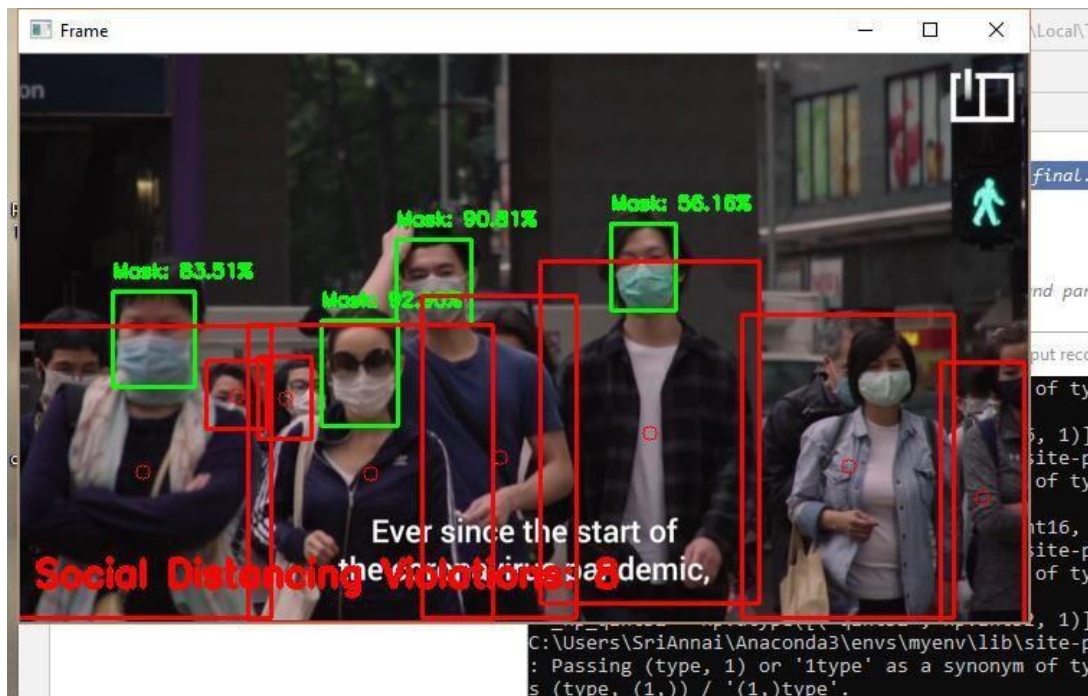
B. SCREEN SHOTS

I. IMAGE INPUT

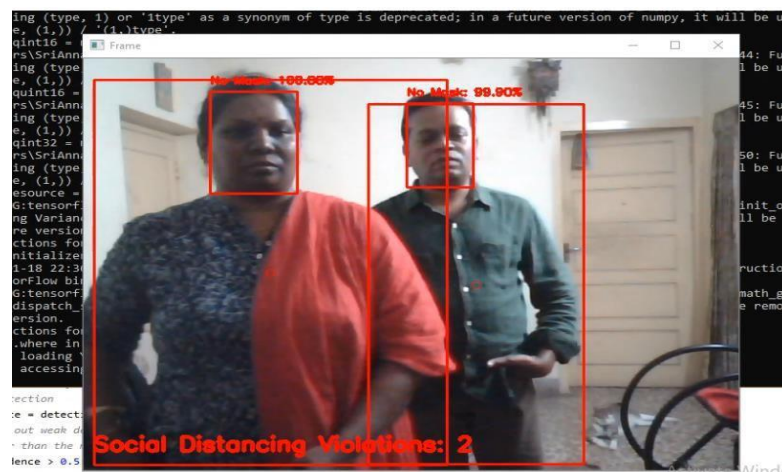
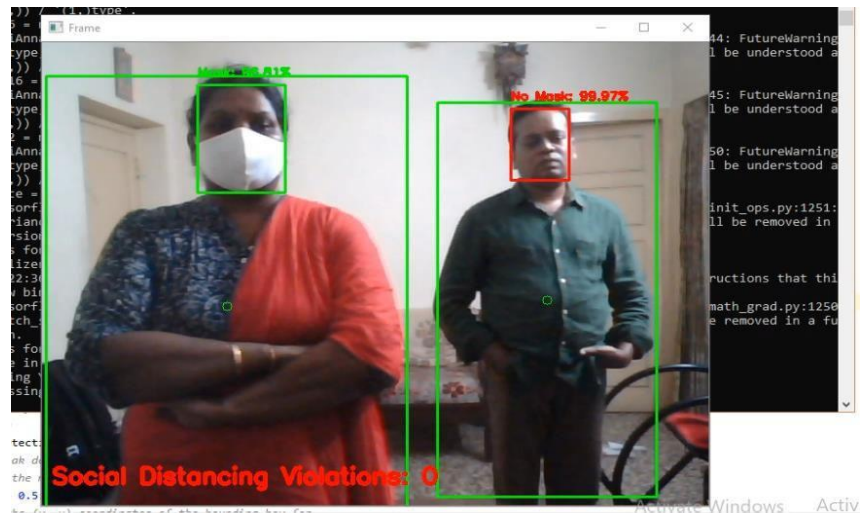




II. VIDEO INPUT



III. LIVE CAMERA INPUT



C. PUBLICATION WITH PLAGIARISM REPORT

I. PAPER

COVID-19 Prevention Melioration : Face Mask and Social Distancing Detection

Aishwarya S
Department of Computer Science and
Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
aishu82ki@gmail.com

Nirmalrani V.
Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
nirmalrani.v@sathyabama.ac.in

Abstract—The pandemic Coronavirus disease 2019 (COVID-19) gave rise to a huge impact on everyone's lives. Millions of people have been infected worldwide. Now the vaccination has been found, WHO has still insisted on wearing mask and maintaining social distance is necessary as it is the most feasible way to prevent ourselves from the virus. This notion motivated me to contribute an efficient computer vision based detection system to public safety. The detection system detects if there is any violations through the camera and on any video or image input. In this system, we have used deep learning algorithm. This system I have proposed here can be applied in places where people gather in huge amount. This automated detection technique makes it easier to inspect the public gathering in malls, streets, any firms, apartments, etc.

Keywords—Computer Vision, Deep Learning, OpenCV, Object detection, Face detection, COVID-19, Face mask, Social distance, Single Shot-Multibox Detector (SSD), Convolution Neural Network (CNN), You Only Look Once (YOLO).

I. INTRODUCTION

Computer vision [9] is the most powerful and compelling types of Artificial Intelligence. The field which is subjected on duplicating the complex things of human vision system and using the computer systems to obtain and execute the objects from images and video like humans do is termed as Computer vision. The way of recognizing the patterns is known as Computer vision. One of the many ways, is where we educate a computer system on how to look out for visual data is to cater thousands or millions of information labeled images and push those feeds into multiple software techniques or algorithms to create or visualize the possible patterns that are most similar to that of the labeled items.

The development of computer vision has made its mark recently in analyzing and solving the complex problems which has the relevant potential to contribute and help the health care industry in controlling COVID-19 [4].

Artificial intelligence (AI) [2] is the replication of human intellect in machines that are made to work similar as humans and replicate the actions. This label can also be viable on any machine that performs same as human mind.

The ideal functionality of artificial intelligence is its capability of taking steps that have the highest chance to achieve the desired goal. Machine learning is the subset of artificial intelligence, which points out that the computer programs spontaneously learns and alter to new details without the human's assistance. Deep learning is a technique which learns by absorbing details from large amounts of unstructured data such as text, images, or video automatically. Algorithms plays a important role in structuring artificial intelligence. The complexity of the algorithm is decides based on the application's complexity.



Fig. 1. Pictorial representation of relationship between Deep Learning, Machine Learning, Artificial intelligence and Computer Vision.

Deep learning is an artificial intelligence method which replicates human brain's working in data processing for object detection, speech recognition, language translation, and decision making. Neural network architectures are used in most of the deep learning methods, therefore deep learning models are mentioned as deep neural networks in most of the cases.

The word "deep" means the number of hidden layers in the neural network. The deep networks can have as many as 150 [18].

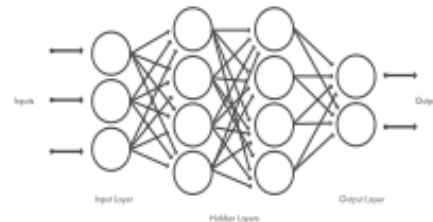


Fig. 2. Layers of neural networks.

Deep learning [17] uses a ranked level of artificial neural networks to perform the process of machine learning. The artificial neural networks is developed like human brain, they are connected like web with neuron nodes. In traditional methods, programs are constructed and to analyse with specified data in linear way, whereas deep learning frameworks makes machines to execute labeled data using nonlinear approach.

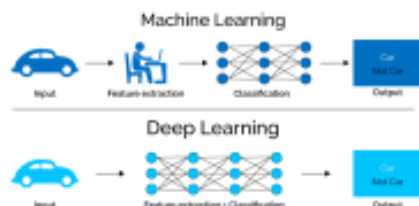


Fig. 3. Difference between the Deep Learning and Machine Learning functionality.

The COVID - 19 [7] has brought down lives of many people around the world [29], [31]. To prevent ourselves from this deadly virus, wearing mask and maintaining social distancing is essential [14]. As it is hard to surveillance every person whether they are not following the precautionary measure, I have proposed a computer vision [7] based system which detects both presence of face mask on people and social distance [19] between people. The system takes input from a live video, images and videos. The system is developed with the combination of various architectures. The ResNet is used for face detection, MobileNetV2 [26] is used as backbone for feature extraction to achieve a best face mask detection model and YOLO [15] is used for object detection [12]. Using the above methods, the system detects the face then looks for the presence of face mask and also compares the distance between people simultaneously. The system thus makes it easier to detect people who are violating the measures in single point of control.

II. RELATED WORK

The face mask and social distancing are the best technique to prevent from the corona virus infection, initially when COVID-19 occurred in end of 2019 the above mentioned ways was brought as an effective measure and it is still has positive impact [25].

Andrew et al [3] have put forward MobileNets which is an efficient models for mobile devices and embedded vision applications. Light weight neural networks are built using MobileNets. These MobileNets are rationalized architecture and it is applied using depth-wise convolutions. Two straight forward global hyper-parameters were introduced by them which are used to efficiently trade between accuracy and latency, so that it grants the model builder the rights to choose the appropriate model for the chosen applications on the basis of the problem's constraints. They have also explained about the potential of MobileNets and effectiveness of MobileNets over various applications. The uses cases of MobileNets includes geo localization, fine grain classification and object detection.

David Oro, Carlos Fernandez, Javier Rodriguez Saeta, Xavier Martorell and Javier Hernandez [8] have presented us a face detector based on Haar that is optimized and which is applied on high definition videos. The operations by kernel mentioned in their paper makes use of both fine grain parallelism and coarse for computation of integral image and evaluation of filter, therefore it is not only helpful for face detection but it is also beneficial for various computer vision implementations.

Wei Bu, Jiangjian Xiao, Chuanhong Zhou, Minmin Yang, Chengbin Peng [30] has come up with a fresh CNN-based cascade framework, that comprises of three thoroughly designed CNN to recognize faces that has a mask on it as there is only less face mask data set that detects accurately, due to which the difficulty to detect face mask is increased. In this paper, they have proposed a fresh dataset named "MASKED FACE dataset" to refine the CNN models and examine the face mask detection algorithm developed on the testing set, and obtain desired performance.

C.Jagadeeswari and M. Uday Theja [5] have highlighted the usage of deep learning technique to track each individual who fail to wear face mask. The mentioned system has been instructed to detect precisely the presence of face mask on a person's face. Once the model finds a person without face mask, people around are alerted with an alarm also the authorities responsible are also alerted, by which appropriate actions can be implied against people who are violating the rules. Their paper have also shown a detailed comparison between the classifiers like VGG16, RESNET50, MobileNetV2, each with optimizer SGD, ADAGRAD and ADAM.

Adithya K and Jismi Babu [1] have developed an efficient face mask monitoring system using machine learning and image processing analysis. They have used convolution neural network which has a high accuracy and decision making.

Kiesha Prem et al. [14] have surveyed on how the mix of population change have affected outbreak in Wuhan. People there have used private location - specified contact patterns and used those ideas in schools, workplace, and reduction of population mixing in general communities. The epidemiological parameters captured and the matrices mentioned above, the latest figures of the Wuhan outbreak is estimated. It has been replicated on the ongoing situations in Wuhan adapting the age-structured susceptible-exposed-infected-removed (SEIR) model for various physical distance measures. They also worked on simulation models to check the dangers and effects of current scenario's outbreak, on how lifting the control measures would affect the people and guiding by allowing them to return to work in specified stages.

Narinder Singh Punj, Sanjay Kumar Sondhadra and Sonali Agarwal [20] in their paper have put forward an automated framework built out of deep learning for monitoring social distance between the individuals by using the surveillance camera videos. The individuals are segregated from the background using YOLO v3 object detection model. The identified individual are tracked with bounding boxed and labels assigned using deep sort approach. An experimental analysis was carried out, from which it was noted that the YOLO v3 framework with deep sort tracking approach displayed desired results with balanced mAP and FPS score to detect social distancing violation in real-time.

Shashi Yadav [27] proposed a computer vision based, live automated monitoring system that detects people's both social distancing and face masks in public area by implementing it on raspberry pi4. After any breach detection, the system sends alert to state police headquarters and also alarms the public. In his system deep learning has been mixed with geometric method for developing a robust

applied to mark the contiguous one for each person that makes certain of the closeness.

- If any person that satisfies closeness property is marked as a neighbour. The contiguous ones are represented in discrete color and the social distancing violation count is incremented. If there is no violation, the count value remains null.

C. Flow of the Proposed System

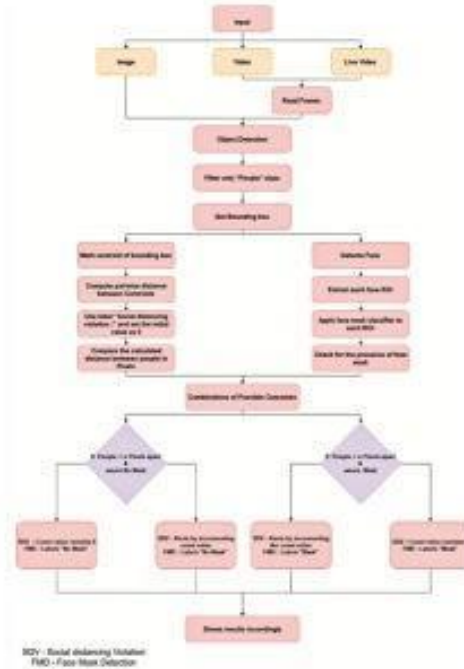


Fig. 6. Flow of the proposed system which is a combination of face mask detection and social distancing detection.

The modules involved in this detection system is discussed previously. Those modules are brought together and made to work as single system. The overall proposed system detects both face mask and social distance violation detection at single point of control.

The proposed system makes it easier to detect the violation and it consumes less time by showing the output of each module simultaneously rather than implementing each module separately.

IV. RESULT AND DISCUSSION

The above discussed COVID - 19 Prevention violation detection system is a deep learning solution that detects humans from the input in fast and efficient manner to measure the distance between people and comprise data collected to create a face mask detection model[24].

In our proposed system we have followed few steps to bring out an efficient and fast detection system.

A. Data collection and data pre-processing

The detection system I have proposed here has a data set that comprises of labeled face images with masks and without mask and they are used for training the model. The data set we use for training the mask detector comprises of nearly 3850 pictures. The datasets are partitioned as training data set and the testing data set initially before labeling it. The training data must have 80% images for training the algorithm accurately and effectively. The testing data set must have 20% images for testing the algorithm's accuracy. Now the trained data collected are classified as Mask and No Mask. The social distancing violation detection is performed by YOLO [11] algorithm.



Fig. 7. Sample pictures from the face mask dataset.

B. Model Training

The Caffe model is used for face detection in our proposed system, which is faster and can process over 60M images in single shot. Once the face has been detected, we need to train a model to detect the presence of face mask on the face. We are using 80% of the data set to train the model with a SSD. Then the data set is stored in the directory and on the basis of the labeled images the algorithm is trained. In pre processing steps, SSD model is given as input to create a model and before that the images in data set are labeled. The model is built using MobileNetV2 as backbone and TensorFlow Object Detection API to train the model.

The social distance detection is done by identifying the person initially. Therefore, I have used YOLO for object detection filtering only person class that is fast and accurate. After the person is identified, we compute the centroids of each person using the bounding box present for each one. Then, the distance between the centroids calculated is measured to predict the distance between each people.



Fig. 8. Graphical representation of the accuracy of the face mask detection model trained.

C. Model Testing

The model is trained with the data set that we gave, the test data set's accuracy is seen by displaying the bounding

modal. The work here detects both with a accuracy of 85% in best environments.

These approaches mentioned above is not sufficient and efficient enough to obtain a full fledged model that helps in public's safety.

III. PROPOSED WORK

The proposed system is an efficient detection system which detects the presence of face mask on each person's face and whether there is social distancing violations. It detects from the live video surveillance, existing images and videos which makes it unique from other existing systems. The system proposed here detects both simultaneously and far more accurately. It ensures public's safety.

This section briefly explains about the models used to automate the detection, the module implementation and the architectural flow of the whole proposed system.

A. Models

The proposed system have been formed using various model. Each of the models were used for specific roles to attain the desired efficient detection system.

- The system needs to detect the faces initially to check for the presence of mask. Therefore, I have implemented face detection using Caffe model. The Caffe model is based on the SSD [28] and it uses ResNet [13] architecture as its backbone. This model can process over 60 million image in single day which makes it easier to train and detect with greater accuracy.
- Once the face detection is done precisely, the face mask detection has to be done. To detect the face mask, we need to apply image classification to classify faces with mask and without mask. This is performed with MobileNetV2. It is a deep CNN [32] with 53 layers. We can use "classify" to classify new set of images using MobileNetV2 model.
- In case of social distance detection, we need to detect the object first. In our case, persons are the object. We have used YOLO [10] for object detection. It processes images in real time at minimum of 45 frames per second and maximum of around 155 frames per second. It has wide margins and it is highly generalizable.

B. Module Implementation

The detection system comprises of two modules. The Face mask detection system is one and Social distancing detection system is another. Let us see the work flow of the modules mentioned above.

Face mask detection system is a separate module which detects the presence of face mask on people's face. The module is broke down into two phases.

1. Training: Here the mask detection data set is loaded from the disk, model is trained on the data set and the mask detector is serialized to the disk.
2. Deployment: Once it is trained, the mask detector is loaded, face detection is performed and it is classified as mask or no mask.

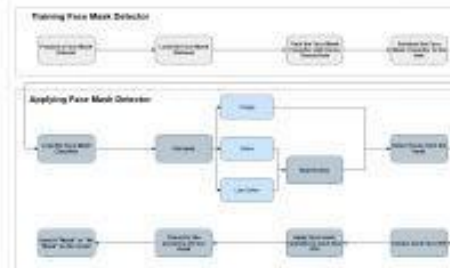


Fig. 4. Flow of the Face mask detection module.

- A face mask dataset is prepared and trained for image classification. The trained classifier is then loaded for detection.
- To detect face mask, we get input from the user in various formats. The frames are read from the video input and the faces are detected from the input.
- The Region Of Interest (ROI) of each face is extracted and the mask classifier is implemented to each ROI.
- The presence of face mask is checked and the results are shown. It labels "Mask" if there is a mask on a person's face and if not, it labels "No Mask".

Social distance detection system is another module which detects the distance between the people and notify if there is a violation.



Fig. 5. Flow of the social distancing detection module.

- The social distance detection system gets input from the surveillance camera [21] or from user either in image or video format. The object detection is applied and "People" class is filtered.
- The model generates a set of bounding box [22] around each human.
- Each individual is correlated with 3-dimensional feature area (x,y,d), where (x,y) refers to the centroid points of the displayed bounding box and d refers to the individual's depth that we see from the surveillance camera or from the video or image that we give as input.

$$d = ((2 * 3.14 * 180) / (w * h * 360) * 1000 + 3),$$

where the bounding box's width is referred as w and its height is referred as h.

- Pairwise L2 norm is estimated for the group of bounding boxes. The L2 norm's dense matrix is

box with label and confidence score. The model I have proposed initially detects all the person in the frame and shows green bounding boxes around each person, when persons violate the social distancing norms the bounding boxes alter it's colour to red for those who are violating. It also increments the violation counts whenever a person breaches the norms. The accuracy of the social distance violation detection system is around 85% to 90%. It varies on a live environment. Similarly, the mask detection is performed by displaying bounding boxes around each persons face that includes labels 'Mask' and 'No Mask' including the confidence score. It also alters the bounding box colour to red from green when there is face mask is not present on the person's face. The average accuracy of the face mask detection model is 90%. It can detect with a accuracy ranging between 90% to 99% in a better angle of surveillance.

D. Model Implementation

The system gets input in three different formats which are image, video and from a live camera. From the input the system monitors and detects the presence and absence of the face mask on the person's face and violation of social distancing among the people. When it detects a person without a face mask it notifies us by labeling it as 'No Mask' and if there is a social distancing violation, it intimates us by incrementing the violation count. Either way it changes the bounding boxes colour to red if there is a violation.

V. OUTPUTS

As the system proposed here can be applied on three different formats of input, the outputs of each kind of input is disclosed here.



Fig. 9. Sample outputs of image inputs given by the user in which the presence of mask and social distancing violation is detected.

The proposed system output for the image input (as shown in Fig. 9) initially process the input image and detects the individual and their face from it. Then the people and their face is confined in the bounding boxes with the confidence score. The colour of the bounding box is changed to red when there is a violation and the count value is increased when there is a social distancing violation.

The first image shows that there is a violation of social distancing and absence of face mask on both the person's faces.

The second image shows that both the persons are wearing a face mask and maintaining social distance between them.

The third image shows that the people are violating social distancing but all are wearing face mask.



Fig. 10. Sample output of recorded video input given by the user in which the presence of mask and social distancing violation is detected.



Fig. 11. Sample outputs of the Live video input given from the Camera in which the presence of mask and social distancing violation is detected.

The outputs for the recorded video input (as shown in Fig. 10) and live video input (as shown in Fig. 11) reads the frames from the video and detects the person and face of the person from it. Then the person's face is confined in the bounding boxes with the confidence score. The bounding boxes change colour to red when there is a violation and the count value is incremented when there is a social distancing violation.

VI. CONCLUSION

In this paper, I have put forward a computer vision based system which ensures people's safety in public places from COVID-19 virus infection. It is done by assisting firms, any organizations or government to surveillance through camera equipped with the proposed system. The system can also be applied on any images or videos.

Thus, the system function in an efficient way in the current scenario where the virus is spreading faster and the vaccine has not yet suggested as the only sufficient way to prevent ourselves from the deadly virus. Therefore, detection of social distancing and presence of face mask violations helps in ensuring human health's safety. The solution I have proposed here saves time and reduces the spread of the coronavirus. This system can be applied on places with huge crowd [23] like malls, stations, companies, streets, etc.

VII. FUTURE WORK

The system has only some features that will help in attaining a complete solution for the current problem. There are other use cases that can be added to derive a more efficient system for the safety of people.

The system proposed here can be developed into a full stack application, that can get the data of each person violating the precautionary measures. The details can be used to send a warning message to the person or fine a minimum amount to let them know the importance of following the measures.

As coughing and sneezing is the cause for spreading of COVID-19 infection, including body gesture to detect coughing or sneezing by a person in a crowded place will help the current situation.

Another symptom of corona virus is body temperature, it is difficult and vulnerable of getting infected to note the temperature of large crowd of people with IR thermometers. This system can be equipped to any thermal screening cameras to detect the body temperature of each person in public areas effectively.

REFERENCES

- [1] Aditya K. Jirj, Babu, "A Review on Face Mask Detection using Convolutional Neural Network", *URASET*, vol. 7, no. 11, pp. 1302-1304, 2020.
- [2] Ake Frankfeld, "Artificial Intelligence (AI)", <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>, Investopedia, 2021.
- [3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNet: Efficient Convolutional Neural Networks for Mobile Vision Applications", *arXiv*, 2017.
- [4] Alexander Latta, Jeanie Bom, Aamir Khan, Douglas Pinto, Sanyam Garg, Subrata Chakraborty, Manoranjan Paul, "COVID-19 Control by computer vision approaches: A Survey", *IEEE*, vol. 8, pp. 2169-2536, 2020.
- [5] C. Jajalec, and M. Uday, "Performance Evaluation of Intelligent Face Mask Detection System with Various Deep Learning Classifiers", *International Journal of Advanced Science and Technology (IJAST)*, vol. 29, no. 11, pp. 3074-3082, 2020.
- [6] Chen S., Zhang C., Dong M., Le J., Rao M., "Using ranking-cnn for age estimates", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [7] Chen Wang, Holly Peter W., Hayden Frederick G., Gao George F., "A novel coronavirus outbreak of global health concern", *The Lancet*, vol. 395, no. 10223, pp. 470-473, 2020.
- [8] David Ou, Carlos Fernández, Javier Rodríguez Soto, Xavier Martorell and Javier Hernandez, "Real-time GPU-based face detection in HD video sequences", *IEEE International Conference on Computer Vision Workshops*, 2011.
- [9] Ego, Mihajlovic, "Everything you ever wanted to know about Computer Vision", <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-bec5a-look-why-it-is-so-awesome-ef58df641e>, 2019.
- [10] J. R. A. Fajali and J. Rejman, "Yolov3: An incremental improvement", *arXiv*, vol. 17, 2018.
- [11] Joseph Redmon, Saeed Divya, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [12] K. A. Joshi and D. G. Thakore, "A survey on moving object detection and tracking in video surveillance system", *International Journal of Soft Computing and Engineering*, vol. 2, no. 3, pp. 44-48, 2012.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.
- [14] K. Prem, Y. Liu, T. W. Russell, A. J. Kucharski, R. M. Eggo, N. Davies, S. Flasche, S. Clifford, C. A. Pearson, J. D. Munday et al., "The effect of control strategies to reduce social mixing on outcomes of the covid-19 epidemic in yunnan, china: a modelling study", *The Lancet Public Health*, 2020.
- [15] L. Hensley, "Social distancing is out, physical distancing is in: how to do it", *Global News-Canada* (27 March 2020), 2020.
- [16] Liu H., Chen Z., Li Z., and Hu W., "An powerful method of pedestrian detection based on YOLOv2", *Mathematical Problems in Engineering*, <https://doi.org/10.1155/2018/3518959>, 2018.
- [17] Marshall Hargrove, "Deep Learning", <https://www.investopedia.com/terms/d/deep-learning.asp>, Investopedia, 2020.
- [18] Math Works, "What is Deep Learning?", <https://www.mathworks.com/discovery/deep-learning.html>, [Online; accessed in 2021].
- [19] Mitrail, I., Jang, T., "Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease", *Emerg Infect Dis*, vol. 26(8) pp. 1740-1748, 2020.
- [20] Navinder Singh Pann, Sanjay Kumar, Sandhu, and Sanyal, "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLOv3 and Deepsort techniques", *arXiv*, 2020.
- [21] P. Dollar, V. Ramanan, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features", *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72, 2005.
- [22] Piss, "Object detection and distance measurement", <https://github.com/paul-piss/Object-Detection-and-Distance-Measurement>, 2020, [Online; accessed 01-March-2020].
- [23] R. Fehel and Y. Moses, "Homography based multiple camera detection and tracking of people in a crowd", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [24] S. J. Pan and Q. Yang, "A Survey of Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010.
- [25] Sanchai Sinha Datta, "Mask-wearing together with social distancing has maximum impact on COVID-19", *News Medical* (20 Jan 2021), 2021.
- [26] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: inverted residuals and linear bottlenecks", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510-4520, 2018.
- [27] Shashi Yadav, "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence", *URASET*, vol. 8, no. 7, 2020.
- [28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector", *Springer-European conference on computer vision*, pp. 21-37, 2016.
- [29] W. H. Organization, "WHO coronavirus (COVID-19)", <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>, 2020, [Online; accessed May 02, 2020].
- [30] Wei Bu, Jiangjin Xiao, Chuhan Zhou, Minmin Yang and Changan "A Cascade Framework for Masked Face Detection", *IEEE 8th International Conference on CTS & RAM*, 2017.
- [31] WHO, "Who director-general's opening remarks at the media briefing on covid-19", <https://www.who.int/dg/speeches/detail/>, 2020, [Online; accessed March 12, 2020].
- [32] Z.-Q. Zhao, P. Zheng, S.-L. Xu, and X. Wu, "Object detection with deep learning: A review", *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212-3232, 2019.

II. PLAGIARISM REPORT

PAPER WORK - COVID - 19 PREVENTION MELIORATION.docx

ORIGINALITY REPORT

10%	8%	6%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijraset.com Internet Source	1%
2	export.arxiv.org Internet Source	1%
3	www.investopedia.com Internet Source	1%
4	B. Sathyabama, Ashutosh Devpura, Mayank Maroti, Rishabh Singh Rajput. "Monitoring Pandemic Precautionary Protocols using Real-time Surveillance and Artificial Intelligence", 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020 Publication	1%
5	Imran Ahmed, Misbah Ahmad, Joel J.P.C. Rodrigues, Gwanggil Jeon, Sadia Din. "A Deep Learning-Based Social Distance Monitoring framework for COVID-19", Sustainable Cities and Society, 2020 Publication	<1%