# FACE MASK DETECTION AND SOCIAL DISTANCE ANALYZER

PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF

**BACHELOR OF COMPUTER APPLICATIONS**

**TO**

**MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)**

**Affiliated to**

**MAHATMA GANDHI UNIVERSITY, KOTTAYAM**

By

**ROSHIN JAMES**

(Reg.No:19UBC151)

GUIDED BY

**Dr. Rajimol A.**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)**

**PEERMADE - 685531**

**MAY, 2022**

# DECLARATION

I, **ROSHIN JAMES [ Reg. No: 19UBC151]** certify that the main project report entitled **"FACE MASK DETECTION AND SOCIAL DISTANCE ANALYZER"** is an authentic work carried by me at **Marian College Kuttikkanam (AUTONOMOUS)**. The matter embodied in this projectwork has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Signature of the Student

**ROSHIN JAMES [ Reg.No 19UBC151 ]**

Date: 07-05-2022

# BONAFIDE CERITIFICATE

This is to certify that this project work entitled **"FACE MASK DETECTION AND SOCIAL DISTANCE ANALYZER"** is a bonafide record of work done by **ROSHIN JAMES [ Reg. No 19UBC151]** at **MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)** in partial fulfillment for the award of **Degree of Bachelor of Computer Applications of Mahatma Gandhi University, Kottayam.**This work has not been submitted elsewhere for the award of any other degree to the best of my knowledge.

Head of the Department & Internal Guide

Dr. Rajimol A.

Dept. of Computer Applications

Marian College Kuttikkanam (Autonomous)

Peermade – 685531

Submitted for the Viva-Voce Examination held on

Department Seal

External Examiner

# ACKNOWLEDGEMENT

"Gratitude is a feeling which is more eloquent than words, more silent than silence." In undertaking this project I needed the direction, assistance and cooperation of various individuals and organizations, which is received in abundance with grace of God, without their unconstrained support, the project could not have been completed. If words are considered as the symbol of approval and token of acknowledgement, then let the following words play the heralding role of expressing my gratitude. I wish to acknowledge my sincere gratitude to Rev Fr. Boby Alex, manager, Marian College, Kuttikkanam and Rev. Dr. Roy Abraham P,principal, Marian College Kuttikkanam (AUTONOMOUS), for all their efforts and administration in educating me in this premier institution. I extend my gratitude to Dr. Rajimol A., Head of the Department of Computer Application as well as my internal project guide, who is a constant source of inspiration and whose advice helped me to complete this project successfully. I extend my sincere gratitude to Mrs. Rosamma K.S and for the fruitful training and technical assistance throughout the project, without which this project wouldnot become a reality. With great enthusiasm I express my gratitude to all the faculty members of Department of Computer Application for their timely help and support. Finally I express my deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project successfully.

Roshin James

# ABSTRACT

# ABSTRACT

The COVID-19 pandemic is causing a global health crisis. Public spaces need to be safeguarded from the adverse effects of this pandemic. Wearing a facemask and maintaining social distance becomes the effective protection solutions adopted by many governments. Manual real-time monitoring of facemask wearing for a large group of people is becoming a difficult task. The goal of this paper is to use deep learning (DL), which has shown excellent results in many real-life applications, to ensure efficient real-time facemask detection and parallelly develop a distance analyzer to detect social distance violations in real time. The proposed approach for face mask is based on two steps. A first step aiming to create a Deep Learning model that is able to detect and locate facemasks and whether they are appropriately worn. Second step that deploys the Deep Learning model at edge computing in order to detect masks in real-time. In this study, we propose to use MobileNetV2 which is a type of convolutional neural network that seeks to perform well on mobile devices, to detect facemask in real-time. Several experiments are conducted and show good performances of the proposed approach. In addition, several comparisons with many state-of-the-art models namely ResNet50, DenseNet, and VGG16 show good performance of the MobileNetV2 in terms of training time and accurac

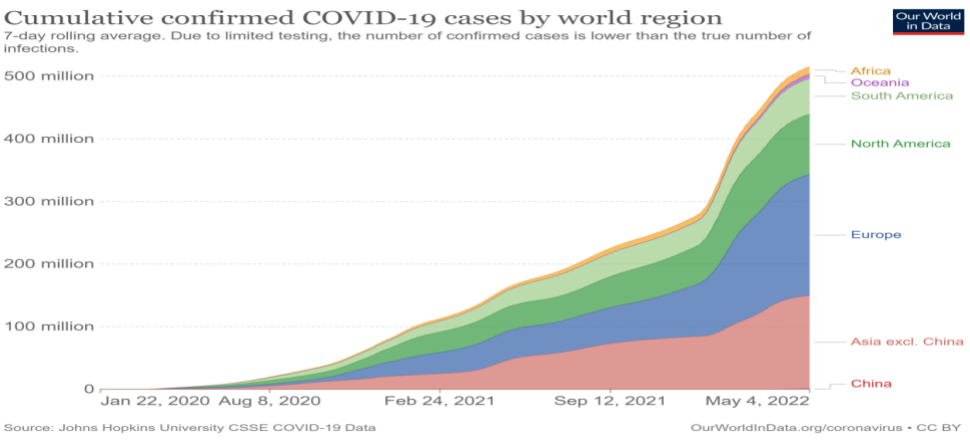
# CONTENTS

# INTRODUCTION

# 1. INTRODUCTION

## 1.1 BACKGROUND

Corona virus disease 2019 (COVID-19) is a contagious disease caused by a virus, the severe acute respiratory syndrome corona virus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China, in December 2019.The disease spread worldwide, leading to the COVID-19 pandemic. COVID-19 transmits when people breathe in air contaminated by droplets and small airborne particles containing the virus. The risk of breathing these in is highest when people are in close proximity, but they can be inhaled over longer distances, particularly indoors. Transmission can also occur if splashed or sprayed with contaminated fluids in the eyes, nose or mouth, and, rarely, via contaminated surfaces. People remain contagious for up to 20 days, and can spread the virus even if they do not develop symptoms. Globally, as of 8:38pm CEST, 4 May 2022, there have been 512,607,587 confirmed cases of COVID-19, including 6,243,038 deaths, reported to WHO. The spread of virus can be avoided by mitigating the effect of the virus in the environment or preventing the virus transfer from person to person by practicing physical distance and wearing face masks. WHO defined physical distancing as keeping at least six feet or two meters distance from others and recommended that keeping the physical distance and wearing a face mask can significantly reduce transmission of the COVID-19 virus. Here, a software system is developed to make face mask wearing and maintaining social distance at public places mandatory. The system here helps to find out those who violates the rules proposed by organizations against covid 19. This system has the capability to capture the images of those who lacks a mask or those who doesn't maintain the distance.



## 1.2 GOAL

The main goal of the this project is to overcome the problem of people not wearing masks and not maintaining social distances in various public places or gatherings. It aims at detecting the faces with mask and without mask and distance between the people using various tools and technologies. Those people who violates mask and distance rules will be correctly captured in the system, as the system implements saving the images of violators. Later these pictures can be used to correctly identify the people who violated the rules. Face mask detection is implemented as a machine learning model in the form of a code written in python language. Social distance analyzer uses object detection followed by distance calculation.

## 1.3 SET-UP

- **Dataset:**

  Here we use dataset only for mask detetction as social distance is based only on distance calculation. Dataset is obtained from online source(kaggle.com) which is of images of .jpg and .png formats. Totally 616 images are there which is divided into two classes-with mask and without mask. With mask contains images of people wearing masks and without mask contains images of people who do not have mask.

- **Features :**

  The system is implemented using python library tkinter, which provides a user interactive GUI through which the user can control the working. The interface contains several buttons through which user can initiate actions. Data preprocessing, training, model creation and model usage are the major steps for face mask detection while social distancing needs the object detection followed by calculation of the distance between the objects. Machine learning algorithms take features of a dataset as input, determine patterns inside it, and create output. The performance of a model depends upon the different types of input variables that we pass to the build model. We construct a Deep learning model using tensorflow, openCV and Keras libraries.

# 2. THEORETICAL BACKGROUND

## 2.1 Machine Learning Basics

Given definitions show the difference between data mining and machine learning, two fields that are built upon each other. Many basics can therefore be considered being data mining techniques, while more advanced topics can be categorized as machine learning techniques. Thus, machine learning and data mining are methods to iteratively detect patterns in data, which is not necessarily structured. This also mean that models are not explicitly programmed with a known result. Examples where machine learning is used in big data are diverse. This includes:

- Fraud detection to find anomalies in tax or credit card data.
- Prediction of so-called "rest of useful lifetime" in industrial machines.
- Text sentiment analysis and opinion mining in social networks such as twitter.
  Financial modeling

### 2.1.1 Data Mining Workflow

The most basic workflow for data mining, and therefore machine learning, can be dividedinto six steps. In the first step data acquisition has to be mentioned, as insufficient or biaseddata can lead to wrong results. In machine learning or data mining this data usually has tobe quite big, as patterns might only emerge with thousands or millions of individual data points. The second and maybe most important step is the cleaning of given data. Problemswith data can include unsuitable features, noisy and so on. Features can be nominal (such as yes/no or male/female), ordinal (ranking such as school grades) or numerical (temperatures, cost and other parameters) and sometimes features have to be converted. Anexample would be to label all days with average temperatures over 10°C as "warm" and allbelow as "cold". Outliers might either be interesting, as in anomaly detection, or can changethe outcome of the learning process negatively, e.g. when using experimental data where outliers have to be removed, therefore this has to be considered in the cleaning process. After all, a scientific or an analytical process can only be as good as the data provided is, as what often jokingly is said: "garbage in causes garbage out".

In the third and fourth step, with the cleaned data and a decision about the chosen modeling technique, is to build and train a model. This is possibly the most abstract step in data mining, especially when pre-built programs such as Azure Machine Learning Studio or scikit-learn (in Python) are used. Simply put, the training process is finding structures in the given training data. What kind of data or features these are is heavily dependent on thegoal (clustering or prediction etc.). Taking a simple example of model training: a very famous dataset is the iris-dataset, that includes features of three different iris-species and uses clustering to determine the correct subspecies. Features included in this set are different parts of the flower, such as petal or sepal length. When using clustering, the training algorithm iteratively calculates the most common features and allows therefore a grouping process of all the data points. The result is heavily dependent on the complexity of these clusters, as is in regression the result dependent on the complexity of a curve. As can be seen in Figure 2.1 it can be quite hard to determine if the bottom-most red data pointbelongs to the species-A-cluster or to the species-B-cluster, depending on the method of measurement.

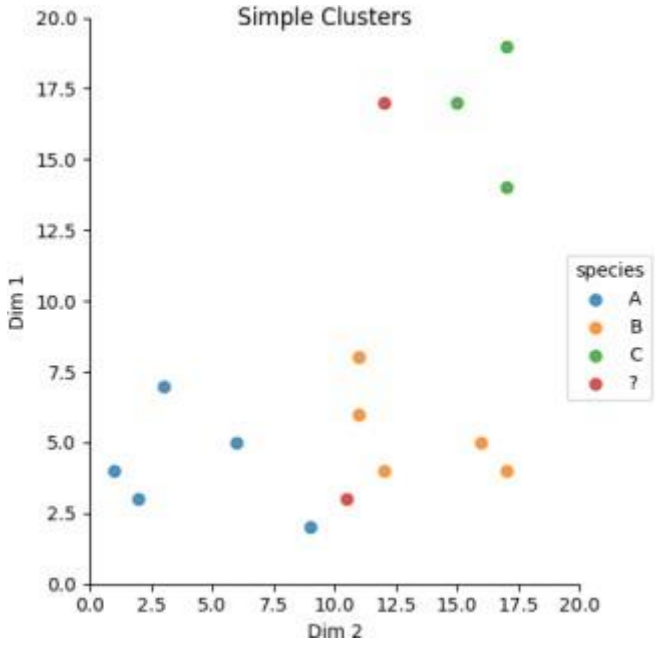



Fig2.1: A simple, made-up, data set of three different species.

Testing and evaluating the model is mostly done by statistical methods and will seldom give a result of 100 % match between the training and validation data. Considering one ofthe most intuitive and simple data mining models, linear regression, this uncertainty is mostly covered by introducing a measurement of uncertainty. A good introduction into machine learning and its workflows can be found in [1]

### 2.1.2 Types of learning:

"Supervised learning is a type of machine learning algorithm that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values.", [2]. Unsupervised learning: "Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input datawithout labeled responses.", [3]. In data mining and machine learning an abundance of models and algorithms can be found, but most fundamentally these are divided into supervised and unsupervised learning. One fundamental example has been mentioned in the foregoing section, the clustering of iris-species. Former is a supervised process where data points are labelled ("species A", "species B" or "species C") and labels are calculatedfor new data points. Comparing calculated labels according to the trained model with the original label gives the model's accuracy, hence supervised. Unsupervised learning on the other hand does not require any labelling, since the algorithm is searching for a pattern in the data. This might be useful when categorizing customers into different groups withouta priori knowledge of which groups they belong to.

One another type of learning is Reinforcement learning that works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving itsperformance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards[4].

### 2.1.3 Machine Learning Algorithm Types

For machine learning many different algorithms can be found. For simplicity these can be subdivided into four categories, where each category is good for different kind of problems.Anomaly detection algorithms, are good for finding unusual data points. Trained classification algorithms can be used to categorize unseen data. As an example, it could beused to take in data from a phone on movement to categorize what activity is being performed. Clustering algorithms group data into clusters and look for the greatest similarities. This can be used to find unknown connections on huge sets of data. Regressionalgorithms are used to find patterns and build models to predict numerical values from datasets. These will take multiple inputs and determine how much each input affects the output.

To reiterate, Machine Learning is simply recognizing patterns in your data to be able to make improvements and intelligent decisions on its own. From the implementation point of view, Python offers concise and readable code. While complex algorithms and versatileworkflows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML probleminstead of focusing on the technical nuances of the language.

Python, with its rich technology stack, has an extensive set of libraries for artificial intelligence and machine learning. Some of them are Keras, TensorFlow, and Scikit-learn for machine learning, NumPy for high-performance scientific computing and analysis, SciPy for advanced computing, Pandas for general-purpose data analysis , Seaborn for data visualization.

### 2.1.4 Bias-Variance-Trade-off

It is important to understand prediction errors (bias and variance) when it comes to accuracy in any machine learning algorithm. There is a tradeoff between a model's abilityto minimize bias and variance which is referred to as the best solution for selecting a value

of Regularization constant. Proper understanding of these errors would help to avoid the overfitting and underfitting of a data set while training the algorithm.

A problem in machine learning is to find a balanced compromise between training accuracyand validation accuracy. This means to find a function that solves a training problem accurately, e.g. a high-degree polynomial that fits the training data well, but is notoverfitting the validation data. On the other hand, a function too simple can oversimplify (underfit) a problem and neglect present patterns.

## 2.2 Neural Networks

In the most basic definition, a neural network is a simulation of an animal (human) nervoussystem based on artificial neurons. Artificial neurons are supposed to imitate the function of a biological neuron; one or more inputs are being summed up and compared to a "firing"threshold that generates an output. The term "to fire" is used in biology and refers to a biological model of neural activity. Neural networks excel at complex tasks such as language recognition, computer translation or image recognition and what makes their useeven more interesting, they are considered to be simply understood and constructed, as thefundamental mathematics behind it is relatively simple. By using the most simple neural network as an example, this section will introduce the basic mathematical concepts behindneural networks. An artificial neuron is, in a simplified sense, a series of inputs. These inputs are weighted and then the products of inputs and weights are summed. Then a threshold comparison decides the output. A single neuron with several inputs and one output is called a "perceptron". In the following example, a made up data set of berry characteristics shall be used to teach a neural network the distinction between lingonberriesand bilberries - the European variant of the blueberry. Characteristics that are measured arecolor, diameter, and weight. For any human - that has been taught to recognize the two berries - this would be a very simple task, since the colors are so clearly different. Nonetheless, an application might be a factory setup with a robot that has to differentiate tons of berries for packaging. Since color recognition is not an option, the computer has todifferentiate between the species by using weight and diameter, that are assumed to be measured quickly and easily. For a berry, x1 and x2 are the inputs to the perceptron. The

classification depends on whether ∑wixi is above or below a threshold value. It can be assumed that if the expression is bigger than threshold it is a lingonberry (or ’1’), otherwiseit is a bilberry (or ’0’). To be able to make this differentiation the neural network has to betaught the weights to find as many correct answers as possible. To teach a neural networka training set is used. In the given example 20 berries have been sorted by a biologist and used for training. Starting with random values for the two weights, w1 and w2, the learningalgorithm will now compare the inputs and (known) outputs of the training set. To find thehighest possible accuracy the weights have to be adjusted step by step. As soon as a maximum of berries are correctly recognized, the model can be seen ready. For validationanother set can be used to check the accuracy on an even bigger scale. Most of the times itcan be seen as beneficial to have very big training sets or to extract outliers and other irregularities. This makes learning in neural networks, or machine learning in general, more difficult, as the data has to be both of high quality and of large quantities. Neural networkscan be used for regression, although, depending on the complexity of the output, it might be excessive. Especially, if the mathematical output function is simple, linear or multiple linear regression might be a better choice, although these tend to easily overshoot functions.Therefore, a different technique is described in the following section. Further informationabout Neural Networks read [5] and [1].

# 3. METHODOLOGY

## 3.1 DATASET REVIEW

In this system we need dataset only for face mask detection as social distancing is based on object detection followed by the calculation of the distance between the correctly identified objects. The dataset used here is obtained from an online source called kaggle.com which allows free download of various dataset. Totally 616 images are there in the dataset which is of jpeg and png formats. It is enclosed with the project under a folder named as Mask dataset. The images are further divided into two classes namely with mask and without mask. With mask contains the images of people wearing face masks properly and without masks contains the images of people who do not have mask.308 images are of with mask type while another 308 images comes under the class of without mask. All of the images are of real people and there is no use of graphically generated images in the dataset. With mask and without mask are the two class labels to which the test data is going to get classified at the time of testing. Binary classification is done in here as there is only two class labels under which the test data is going to get predicted. Pixel arrays of images are created at data preprocessing stage using numpy library for further processing, and hence there is no usage of csv files regarding the dataset. Example of the images in dataset are as follows:

**With mask:**

**Without mask:**

## 3.2 SOFTWARE INTRODUCTION

### 3.2.1 Python 3.10:

Python 3.10.4 is the newest major release of the Python programming language, and it contains many new features and optimizations. This is a special release that fixes a regression which caused Python to no longer build on Red Hat Enterprise Linux 6. There are only 10 other bugfixes on top of 3.10.3 in this release. Improvements in type annotation, structural pattern matching, better error messages, and performance.

o Type Guards: Conditional Type Narrowing
o Parameter Specification Variables: Type Annotations for Decorators
o Explicit Type Aliases
o Structural Pattern Matching
o Better Error Messages

### 3.2.2 OpenCV:

Open CV (Open-Source Computer Vision Library) is a open source computer vision software library for the purpose of machine learning. Open CV was developed to serve the purpose of computer vision applications and to stimulate the usage of machine perception in the commercially viable products. Open CV is a BSD- licensed product which is easy for the utilization and modification of the code. The library contains more than 2500 advanced algorithms including an extensive set of both typical and state-of-the-art computer vision and machine learning algorithms. These algorithms can be employed for the detection and recognition of faces, identification of objects, extraction of 3 D models of objects, production of 3 D point clouds from stereo cameras, stitching images together for production of a high resolution image of an entire scene, finding similar images from an image database, removing red eyes from images taken using flash, following eye movements, recognition of scenery and establishing markers to overlay it with intensified reality etc.

### 3.2.3 Tensorflow:

Tensorflow is a free and open-source software library for machine learning. TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow was developed by the Google Brain team for internal Google use in research and production. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow serves as the core platform and library for machine learning. TensorFlow's APIs use Keras to allow users to make their own machine learning models. In addition to building and training their model, TensorFlow can also help load the data to train the model, and deploy it using TensorFlow Serving.

### 3.2.4 Tkinter:

Tkinter is a graphical user interface (GUI) module for Python, you can make desktop apps with Python. You can make windows, buttons, show text and images amongst other things. Tk and Tkinter apps can run on most Unix platforms. This also works on Windows and Mac OS X. The module Tkinter is an interface to the Tk GUI toolkit. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

### 3.2.5 Keras :

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation

and developer guides. Keras makes it easier to run new experiments, it empowers you to try more ideas than your competition, faster.

### 3.2.6 Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib. Several toolkits are available which extend Matplotlib functionality. Matplotlib serves as a visualization utility which is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility. Matplotlib take care of the creation of inbuilt defaults like Figure and Axes.

### 3.2.7 NumPy:

NumPy is an open source library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars**.**

### 3.2.8 imutils:

A series of convenience functions to make basic image processing functions suchas translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV. A series of convenience functions to make basic

image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges.

### 3.2.9 SciPy:

SciPy, a scientific library for Python is an open source, BSD-licensed library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that, it should work with NumPy arrays. It provides many user-friendly and efficient numerical practices such as routines for numerical integration and optimization. This is an introductory tutorial, which covers the fundamentals of SciPy and describes how to deal with its various modules.

## 3.3 RELATED WORKS

There are only few researches related to Face Mask Detection as it is related to a problem that has arrived recently. Here is the survey of some related work done in this field of study previously:

### 3.3.1 Model on Raspberry-Pi

This uses deep learning technique to distinguish facial recognition and recognize if the person is wearing a facemask or not. Dataset collected contains 25000 images using 224-224 pixel resolution and achieved an accuracy rate of 96% as to the performance of the trained model.The system develops a Raspberry-Pi based real time facemask recognition that alarms and captures the facial image if the person detected is not wearing a facemask.

### 3.3.2 Support Vector Machine(SVM) based model

A hybrid model using deep and classical machine learning for face mask detection was developed here. The proposed model consists of two components. The first component is designed for feature extraction using Resnet50. While the second component is designed for the classification process of face masks using decision trees, Support Vector Machine (SVM), and ensemble algorithm. Three face masked datasets have been selected for investigation. The Three datasets are the Real-World Masked Face Dataset (RMFD), the Simulated Masked Face Dataset (SMFD), and the Labeled Faces in the Wild (LFW). The SVM classifier achieved 99.64% testing accuracy in RMFD. In SMFD, it achieved 99.49%, while in LFW, it achieved 100% testing accuracy.

**Support Vector Machine:** Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

### 3.3.3 ResNet-50 based model

In this paper, the proposed model consists of two components. The first component is designed for the feature extraction process based on the ResNet-50 deep transfer learning model. While the second component is designed for the detection of medical face masks based on YOLO v2. Two medical face masks datasets have been combined in one dataset to be investigated through this research. The achieved results concluded that the adam optimizer achieved the highest average precision percentage of 81% as a detector.

## 3.4 PROCESS FLOW

## 3.4.1 FACE MASK DETECTION

In order to create a custom face mask detector and detect mask presence, we need to break the system into two distinct phases, each with its own respective sub-steps: Phase 1 is **Training Model-** Here we'll focus on loading our face mask detection dataset from disk, training a model (using scikit-learn library) on this dataset, and then serializing the trained face mask detector to disk . Then phase 2 is **Deployment-** Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with mask or without mask. After the training of data the model is tested on a small quantity of testing set of data and evaluated on various parameters including accuracy, precision, recall and f1 score to analyze the accuracy of model before deployment. The trained model predicts and classifies each frame as 'with mask' and 'without mask'. The output is printed on each frame, all together making it visible as predicted text output in live stream.

### 3.4.1.1 Model Training

In the machine learning world, model training refers to the process of allowing a machine learning algorithm to automatically learn patterns based on data**.** The trained mask detector model is serialized to the disk as a model file. From the available dataset 80% of the images are used for model training and the remaining 20% of images are used for testing and validation.

#### 3.4.1.1.1 Data Preprocessing

Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning. It is mainly done to remove inconsistencies, errors, noise and outliers before the data is used up for processing.

Here, in this system two python empty lists are created namely data[] and label[] to hold the details regarding the preprocessed data. Data list is appended with the pixel value arrays of the images in the training dataset. These arrays are created with the help of numpy library. And label list is appended with the class label prediction for each image in the dataset i.e, with mask or without mask labels. At first the directory of images with mask is looped and the lists are appended with the necessary data. Similarly, the directory of images without mask is looped and required data is appended to the lists. Later the alphabetic contents of label list is also converted to binary values using sklearn methods. The images are loaded and resized to a fixed shape of (224,224) uniformly. Keras library is mainly used in the step of data preprocessing.

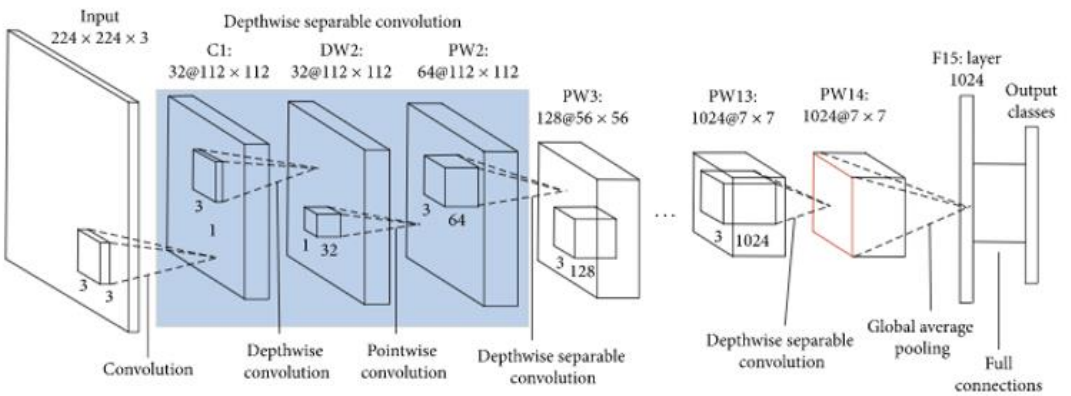
#### 3.4.1.1.2 Training

In this system we are using a type of Convolutional Neural Network(CNN) called as MobileNet V2. Convolution Neural Network (CNN) are particularly useful for spatial data analysis, image recognition, computer vision, natural language processing, signal processing and variety of other different purposes. They are biologically motivated by functioning of neurons in visual cortex to a visual stimuli. What makes CNN much more powerful compared to the other feedback forward networks for image recognition is the fact that they do not require as much human intervention and parameters as some of the other networks such as MLP do. This is primarily driven by the fact that CNNs have neurons arranged in three dimensions. CNNs make all of this magic happen by taking a set of input and passing it on to one or more of following main hidden layers in a network to generate an output.

- Convolution Layers
- Pooling Layers
- Fully Connected Layer

## MobileNet V2

MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobile devices. MobileNetV2s are small, low latency, low power models parameterised to meet the resource constraints of a variety of use cases. MobileNetV2 improves the state-of-the-art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. It is a very effective feature extractor for object detection and segmentation. It is used for depth- wise separable convolutions as efficient building blocks. MobileNetV2 has two main features: the first is Linear bottlenecks between the layers and the second is Shortcut connections between the bottlenecks.The bottlenecks of the MobileNetV2 encode the intermediate input and outputs while the inner layer encapsulates the model's ability to transform from lower level concepts such as pixels to higher level descriptors such as image categories. It gives better accuracy compared to other methods. Below is a diagram depicting the general working structure of the MobileNet V2.




**MobileNet-V2 Architecture**

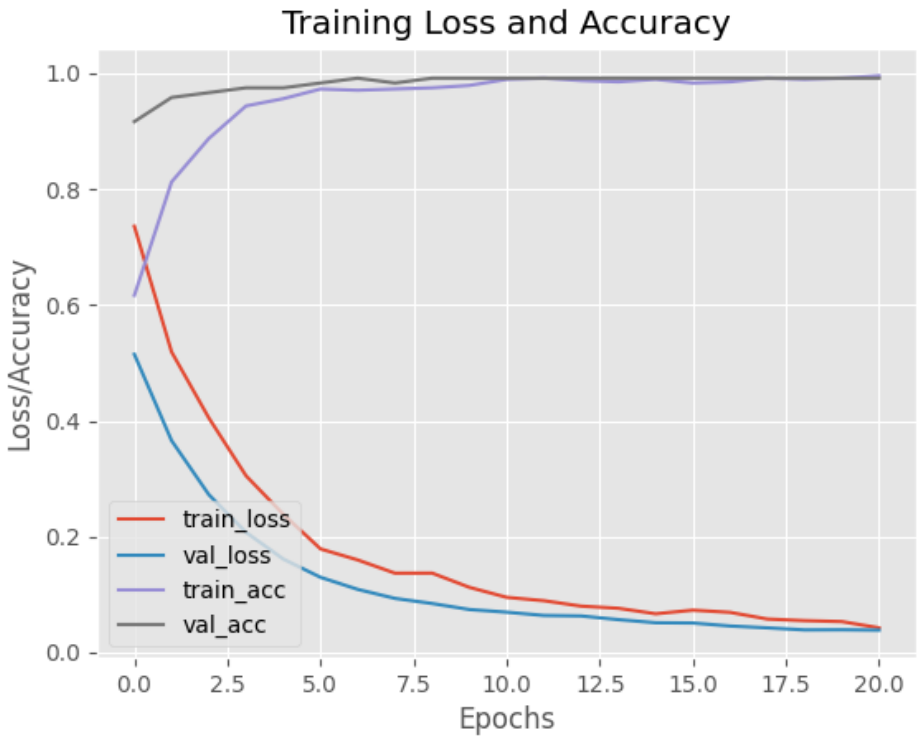
Chiung-Yu Chen

## Testing Details

We perform the Testing of trained model using the twenty percent of the available dataset after successful training of the model. Below image shows out the details regarding the testing of trained mask detector model when the Epochs are given as 21.

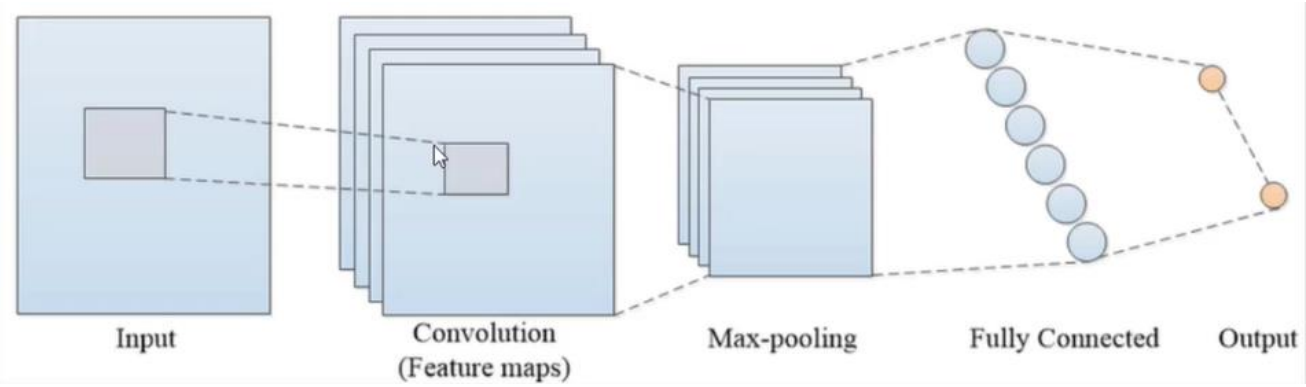
```
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       0.98      1.00      0.99        60
without_mask       1.00      0.98      0.99        60

    accuracy                           0.99       120
   macro avg       0.99      0.99      0.99       120
weighted avg       0.99      0.99      0.99       120
```

## Graphical Details (when Epochs=21)

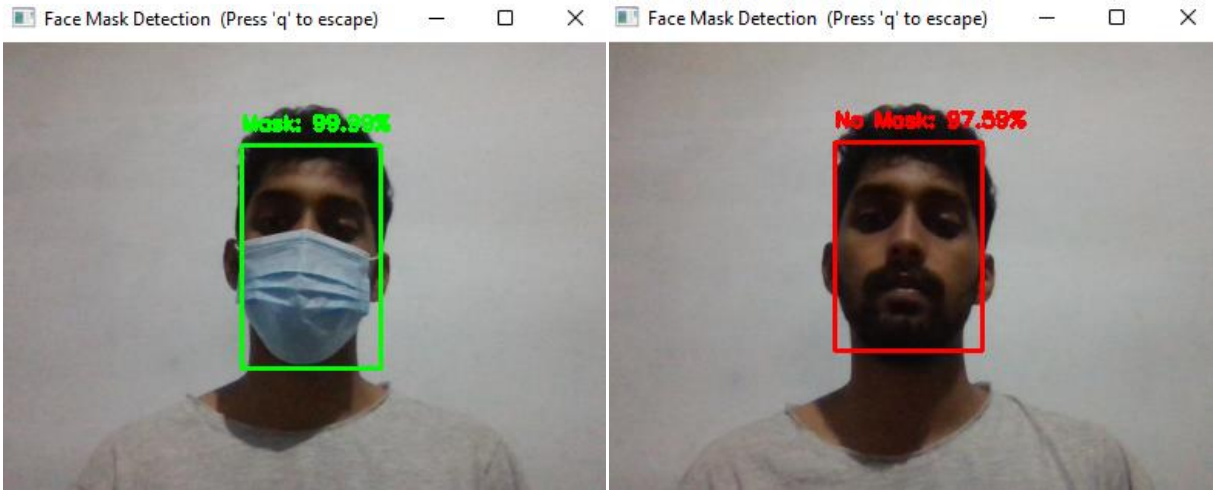


### 3.4.1.2 Deployment

Once the face mask detector is trained, we then move on to loading the mask detector from the disk, performing face detection, and then classifying each face as with mask or without mask on the live video stream. Mobilenet works in the second layer ie convolution in the above working diagram

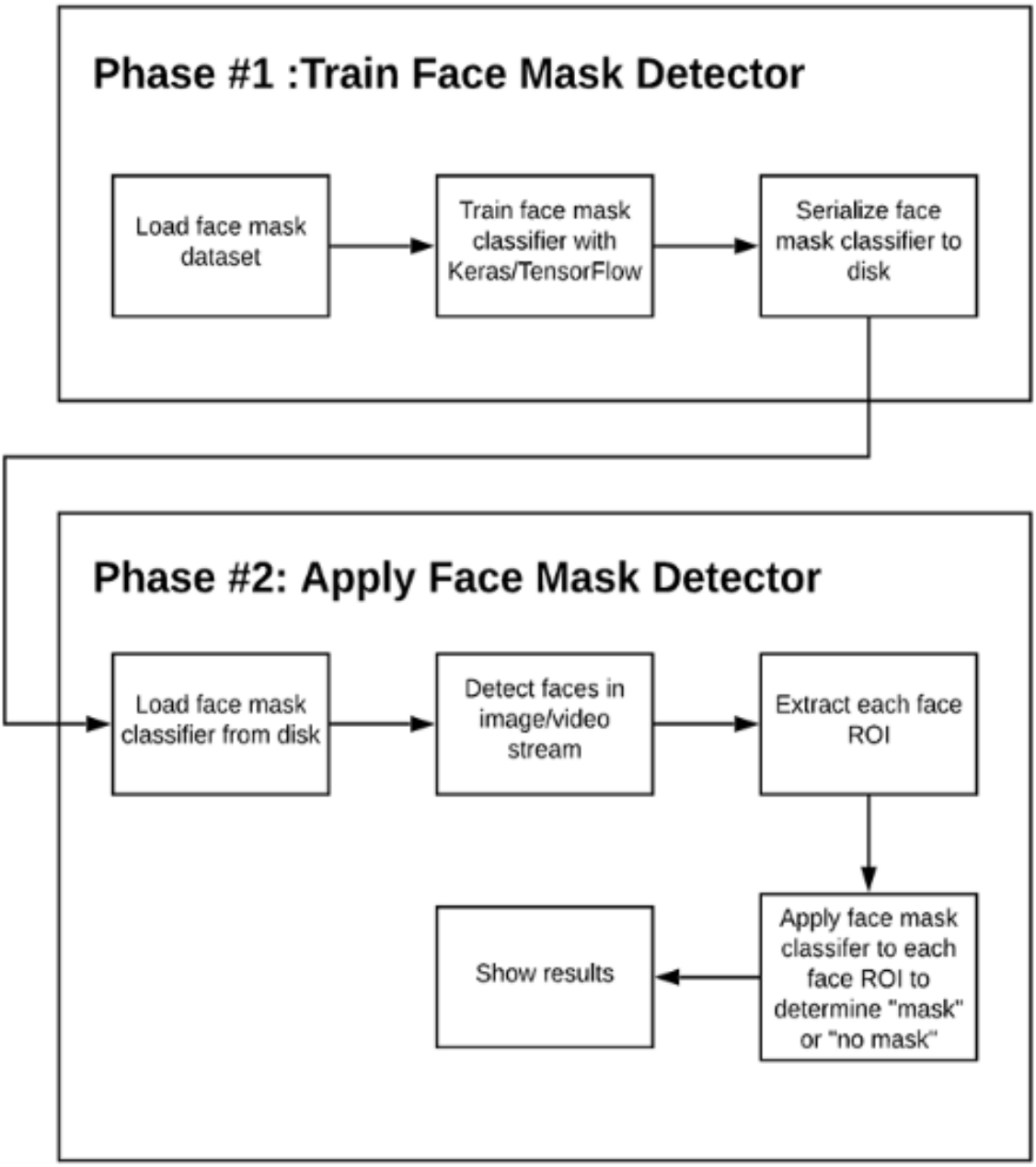



The serialized mask detector model obtained from training in disk is loaded to perform prediction on the testing data. Testing data comes as in the form of frames from the live video stream. Mask detector model detects the presence of face mask mask in each of the frames available and draws conclusion that whether the person in the frame is having a face mask or not. Output of mask detection is as follows.




With Face Mask Without Face mask

## Workflow diagram of face Mask detection



## 3.4.2 SOCIAL DISTANCE ANALYZER

In social distance analyzer ,at first an object detection technique is used todetect objects. Here we detect faces as the objects from the live video frames.Then a square box is drawn around all the detected faces in the frame and the centroid of each of the square boxes are calculated individually. Then the distance in space between the centres of each of the boxes are calculated. There will be a user specified threshold value to classify people as violators or not based on the distance between centre of boxes. If the distance value calculated falls below the threshold value then it is considered as a violation and the persons within the violated box are identified as the distance violators. Then the images of violators are stored in the system for further actions. An alarm will go live if any kind of social distance violations are found in the live video stream.

The steps to build a social distancing detector include:

i. Apply object detection to detect all people (and only people) in a video stream.
ii. Compute the pairwise distances between all detected people.
iii. Based on these distances, check the social distancing and show results.

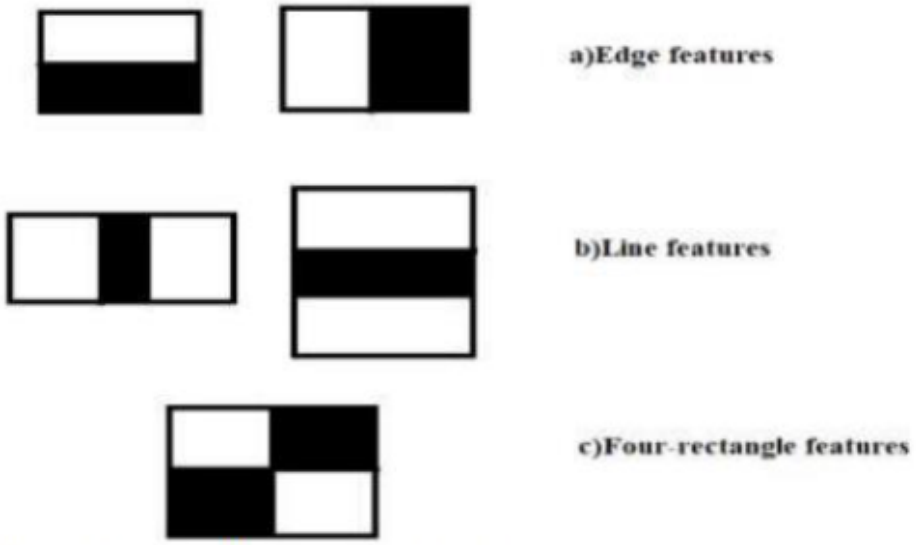
### 3.4.2.1 Object detection (People detection)

In social distance analyzer first we have to detect the faces to measure the distances. Objects here is considered as the people. We use Haar Cascade Classifier for effective object detection from the live video streamed frames.

## Haar Cascade Classifier

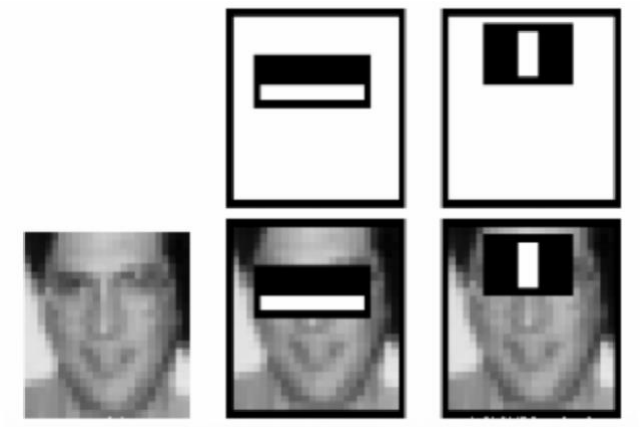
Detecting objects with the help of Haar cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. Object Detection comes under machine learning based approach where a cascade function is trained from lots of positive and negative images. Now what are these positive and negative images? A classifier (namely cascade of boosted classifiers working with haar like features) which is trained with many samples of a specific object (i.e., a face or a car), called positive example. So, whatever you want to detect if you train your classifier with those kinds of

values. For example, if you want to detect face then you need to train your classifier with number of images which contain faces. So, these are called positive images which contain the object which you want to detect. Similarly, we want to train the classifier with negative images that means the images which doesn't contain object that you want to detect. For example, if we want to detect the face then the image which doesn't contain the face is called negative image. In the same way if the image contains face or number of faces then it is called positive images. After a classifier is trained it can be applied to the region of interest in an input image and classifier outputs 1 if the region is likely to show the object or 0 otherwise. Here we will work with face detection. Initially, in order to train the classifier, the cascade function needs a lot of positive images (images which contains faces) and negative images (images without faces). Then we need to extract features from it. For this, we use Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is claimed to be one value which is obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.




Now to calculate lots of features, all possible sizes and locations of each kernel are used. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). In order to calculate each feature, we need to find the sum of the pixels under white and black rectangles. To get over from it, they introduced the integral image. Calculation depends upon the

size of the image if how large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. It makes things superfast. But among all these features most of them are irrelevant that we calculated. For example, consider the image below. The top row shows two good features. In the first feature it focuses on the region of the eyes which is commonly darker than the region of the nose and cheeks. When comes to the second feature it focuses on the property that the eyes are often darker than the bridge of the nose. But if it is applied to cheeks or any other place is irrelevant that you can observe in the image. By using Ad boost we select the best features out of 160000+ features.



In the same way, we have to apply each and every feature on all the training images. It finds the best threshold for each and every feature which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We only select the features with minimum error rate because they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each and every image is given an equal weight in the beginning. After each classification, there will be a change in weights in which weights of misclassified images are increased. Then the same process is done again. New error rates and new weights are calculated. The process will be continued until the required accuracy or error rate is achieved or the required number of features is found). The final classifier is obtained by weighted sum of these weak classifiers. It is then called weak classifier because it alone can't classify the image, but together with others forms a strong classifier.

### 3.4.2.2 Distance Calculation

The faces are detected first using the object detection method, then a box is drawn around the face and the centre of the box is calculated using the diagonal coordinates. If the coordinates of the diagonal of the box are as follows (X1,Y1) and (X2,Y2) then the centre of the box can be calculated by using the equation as follows.

$$Xmid=(X1+X2)/2$$

$$Ymid=(Y1+Y2)/2$$

(Xmid,Ymid) gives the centre of the box drawn around the face detected. Similarly we find the centre of every face detected in the frame. And then the distance between the centers are calculated to check the distance. We use Euclidean distance measure to calculate the social distances. And the result is shown based on the Euclidean distance calculated.

## EUCLIDEAN DISTANCE

Euclidean Distance is a distance between two points in the plane that measures the length of a segment connecting the two points. It is the general way of representing the distance between two points. We use Pythagorean Theorem to calculate the distance between two points as per Euclidean measure

The Euclidean distance formula says:

$$\mathbf{d = \sqrt{[ (x2 - x1)^2 + (y2 - y1)^2]}}$$

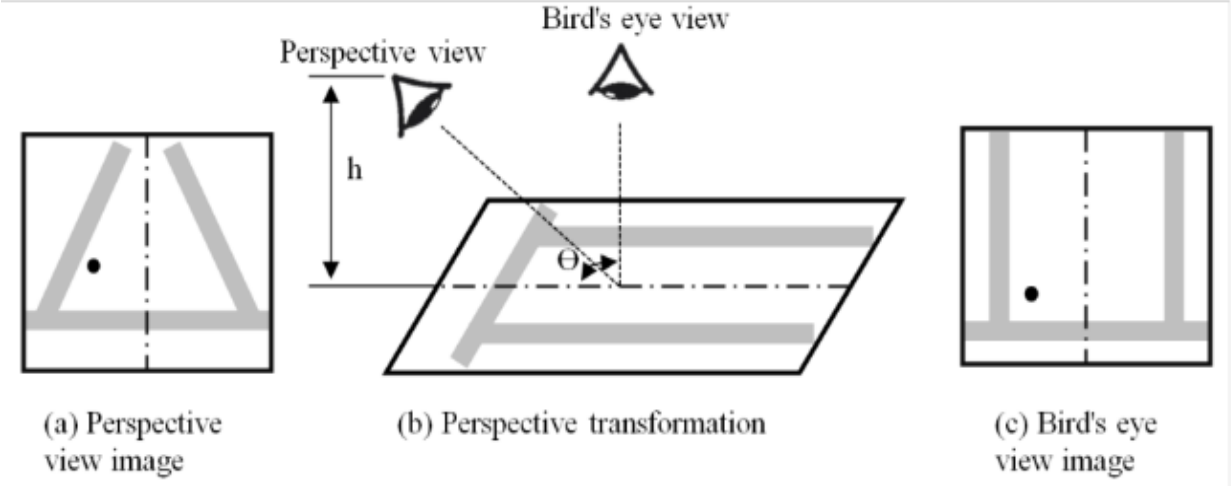
where,

(x1, y1 ) are the coordinates of one point.

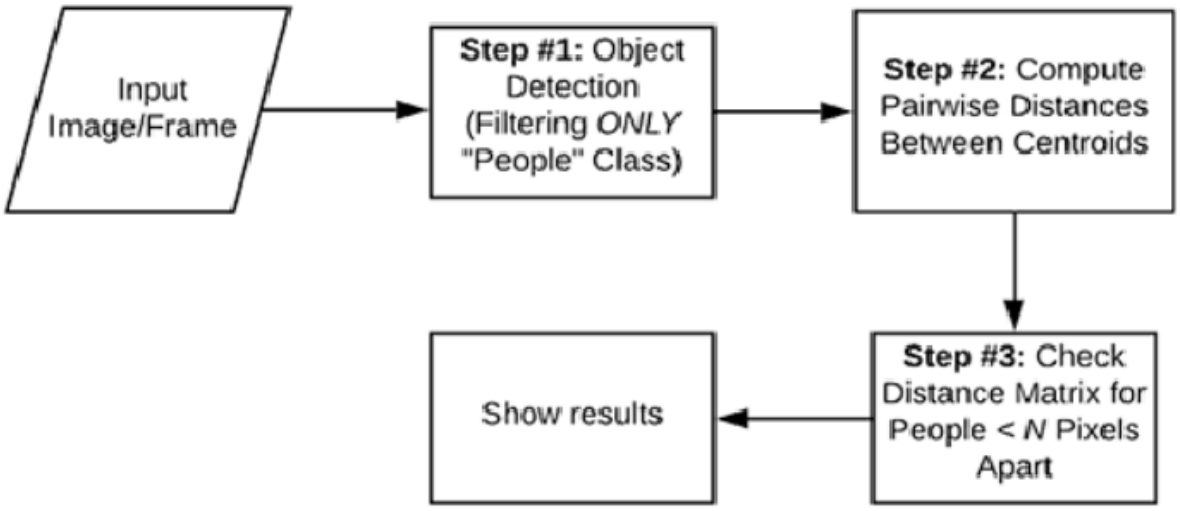
(x2, y2) are the coordinates of the other point.

d is the distance between (x1, y1) and (x2, y2)..

The picture transformation was implemented to project the pictures captured by the camera from an arbitrary angle to the bird's eye view. Underlying figure shows the original image captured from a perspective to the vertical view of bird's eye, where the dimensions in the picture have a linear relationship with real dimensions.




## Workflow Diagram of Social Distance Analyzer



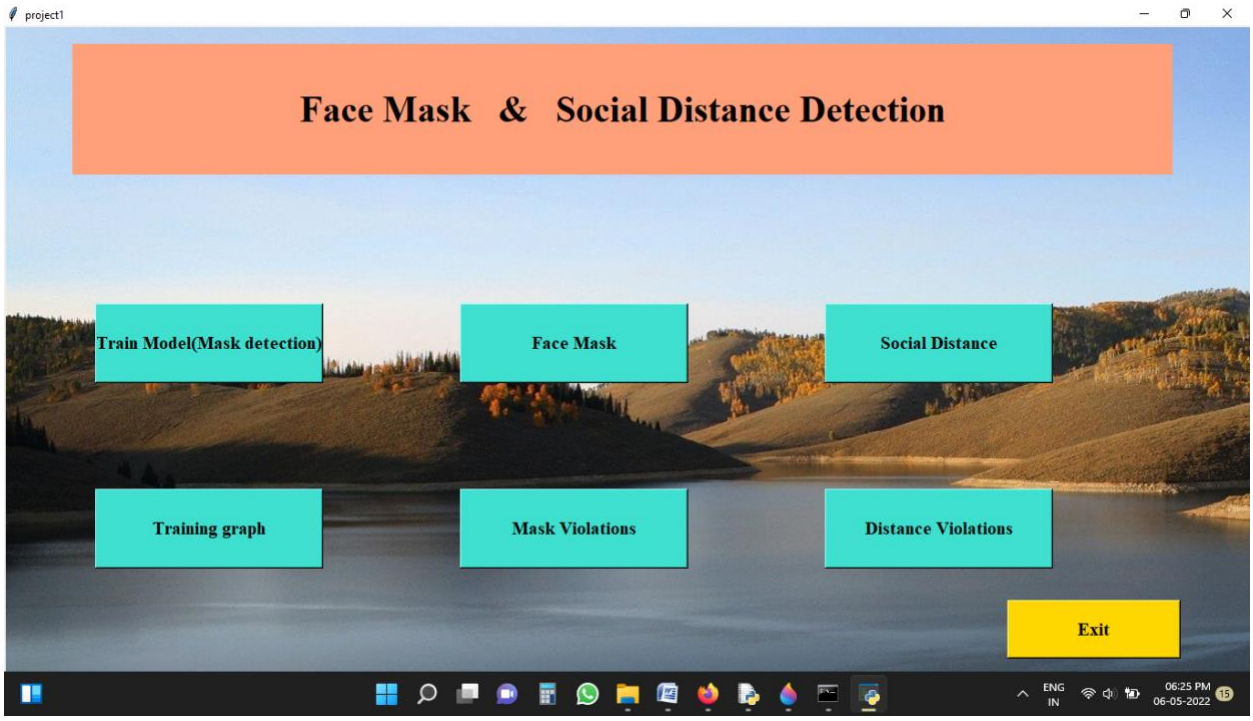
# CONCLUSION AND FUTURE SCOPE

In this paper, a face mask detection and social distance analyzer system was presented which was able to detect face masks and distance violations. The architecture of the system consists of Keras /Tensorflow and OpenCV. A novel approach was presented which gave us high accuracy. The dataset that was used for training the face mask model consists of more than 600 face images. The model was tested with real-time video streams. The training accuracy of the model was around 98%. This model is ready to use in real world use cases and to implement in CCTV and other devices as well. Deploying our face mask detector and distance analyzer to embedded devices could reduce the cost of manufacturing such face mask detection and distance detection systems, hence why we choose to use this architecture.

Our social distancing detector did not leverage a proper camera calibration, meaning that we could not (easily) map distances in pixels to actual measurable units (i.e., meters, feet, etc.). Therefore, the first step to improving our social distancing detector is to utilize a proper camera calibration. Doing so will yield better results and enable you to compute actual measurable units (rather than pixels). For face mask detection, combining an object detector with a dedicated with mask class will allow improvement of the model in two respects. First, the object detector will be able to naturally detect people wearing masks that otherwise would have been impossible for the face detector to detect due to too much of the face being obscured. Secondly, this approach reduces our computer vision pipeline to a single step rather than applying face detection and then our face mask detector model, all we need to do is apply the object detector to give us bounding boxes for people both with mask and without mask in a single forward pass of the network.

This system has great scope as the present day is trying to overcome the problems of corona virus . When installed on CCTV cameras will help organizations to find and take actions against those who violates the anti covid rules proposed by the governments. This is a software project but for further applications in real world scenario it can be integrated with certain hardware equipment and can be installed in various public places for surveillance of people wearing face mask or not wearing face masks and social distance maintaining.

# Screenshot


project1
Face Mask & Social Distance Detection
Train Model(Mask detection)
Face Mask
Social Distance
Training graph
Mask Violations
Distance Violations
Exit
ENG
IN
06:25 PM
06-05-2022

# REFERENCES


[1] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques for Building Intelligent Systems. 2017.

[2] 'Supervised learning', Mathworks, [Online]

Available: https://se.mathworks.com/discovery/supervised-learning.html (accessed January 31,2018).

[3] 'Unsupervised learning', Mathworks, [Online]

Available: https://se.mathworks.com/discovery/unsupervised-learning.html (accessed January 31, 2018).

[4] Csaba Szepesvári, University of Alberta, Algorithms for Reinforcement Learning

[5] 'Overview of artificial neural networks and its applications', Xenonstack, [Online]. Available: https://www.xenonstack.com/blog/data-science/overviewof-artificial-neural-networks-and-its-applications (accessed January 31, 2018).