# ABSTRACT

After the new Coronavirus disease (COVID-19) case spread rapidly in Wuhan-China in December 2019, World Health Organization (WHO) confirmed that this is a dangerous virus which can be spreading from humans to humans through droplets and airborne. As for the prevention, wearing a face mask is essentials while going outside or meeting to others. However, some irresponsible people refuse to wear face mask with so many excuses. Moreover, developing the face mask detector is very crucial in this case. This project aims to develop the face mask detector which is able to detect any kinds of face mask and advise them to wear the mask. we will build a real-time system to detect whether the person is wearing a mask or not. We have trained the model using Kera's with network architecture. Training the model is the first part of this project and testing using webcam using OpenCV is the second part. If it detects that the person is not wearing the mask it advices, the person to wear the mask by means of audio message. This algorithm can efficiently detect the face mask and advice the person to wear the mask, if he is not wearing the mask. The main Objective of this project is to detect face mask and advise the person to wear mask.

# CONTENTS


ACKNOWLEDGEMENT I

ABSTRACT III

LIST OF FIGURES VIII

LIST OF TABLES X

**1.** INTRODUCTION

1.1. PROBLEM DEFINITION 1

1.2. PROJECT PURPOSE 1

1.3. PROJECTFEATURES 2

**2.** LITERATURESURVEY

2.1. MACHINELEARNING 4

2.1.1. FEATURES OF MACHINELEARNING 5

2.2. EXISTING SYSTEM 6

2.3. PROPOSED SYSTEM 7

2.4. SOFTWAREDESCRIPTION 7

2.4.1. PYTHON 7

2.4.2. BENEFITS OF PYTHON 8

2.5 DESCRIPTION OF THE PROJECT 8

2.6 DESCRIPTION OF THE LIBRARIES 10

**3**. REQUIREMENTANALYSIS AND TECHNICAL DESCRIPTION

3.1 REQUIREMENTS ANALYSIS 12

3.1.1 FUNCTIONALREQUIREMENTS

3.1.2. NON-FUNCTIONALREQUIREMENTS 12

3.1.3. HARDWAREREQUIREMENTS 14

3.1.4. SOFTWAREREQUIREMENTS 14

3.2 TECHNICAL DESCRIPTION 14

4.SYSTEM DESIGN

4.1. SYSTEMARCHITECTURE 20

4.2. DATA FLOW DIAGRAM 21

4.3 CLASSDIAGRAM 22

4.4 COMPONENT DIAGRAM 25

4.5 SEQUENCEDIAGRAM 26

5. IMPLEMENTATION

5.1 CNN 30

6. TESTING

6.1 UNITTESTING 42

6.2 INTEGRATIONTESTING 42

6.3 VALIDATIONTESTING 42

6.4 SYSTEMTESTING 43

6.5 TESTING OF INITIALIZATION ANDUICOMPONENTS 43

7. SCREEN SHOTS 44

8. CONCLUSION AND FUTUREENHANCEMENT

8.1 CONCLUSION 47

8.2 FUTUREENHANCEMENT 48

9. REFERENCES 49

# LIST OF FIGURES


| Figure No | Figure Name | Page No |
|---|---|---|
| 2.1.1.1 | Traditional Programming vs Machine Learning | 6 |
| 2.1.1.2 | Machine Learning Model | 6 |
| 4.1 | System Architecture | 20 |
| 4.2 | Data Flow Diagram | 21 |
| 4.3 | Class Diagram | 22 |
| 4.4 | Component Diagram | 25 |
| 4.5 | Sequence Diagram | 26 |
| 6.1 | The testing process | 43 |
| 7.1 | User Details Page | 44 |
| 7.2 | User Dataset | 45 |
| 7.3 | Classification Result1 Page | 45 |
| 7.4 | Classification Result2 Page | 45 |
| 7.5 | Classification Result3 Page | 46 |
| 7.6 | Classification Result4 Page | 46 |


# LIST OF TABLES


| Table No | Table Name | Page No |
|---|---|---|
| 6.1 | Test Case for Prediction Result | 43 |
| 8.1 | Table of Accuracy Scores of Various Models | 47 |

# CHAPTER 1
# INTRODUCTION

COVID-19 transmits when people breathe in air contaminated by droplets and small airborne particles. The risk of breathing these in is highest when people are nearby, but they can be inhaled over longer distances, particularly indoors. Transmission is more likely when people are physically close. Masks should be used as part of a comprehensive strategy of measures to suppress transmission and save lives. As masks become a new standard in daily life, it is necessary to be always vigilant to create a safe environment conducive to public safety. Some irresponsible people refuse to wear face masks with so many excuses like negligence and lack of awareness. For security reasons, many areas of society seem to be using some covid tracking tools. One of the most important tools is the mask detector. Generally, in some places humans are appointed to inspect people. This process may suffer from some human errors and consume time. The manual inspection is also extremely dangerous for inspectors as well as the people due to the risk of transmission of COVID.In this Project we will develop a machine learning project – Real-time Face Mask Detection with Python. This project can efficiently detect the face mask more accurately and with no risk of transmission.

## 1.1 PROJECT DEFINITION

Now a days in health industry there are various problems related to machines or devices which will give wrong or unaccepted results, so to avoid those results and get the correct and desired results we are building a program or project which will give the accurate classifications based on information provide don the datasets that are available in that machine. The health industry in information yet and knowledge poor and this industry is very vast industry which has lot of work to be done. so, with the help of all those algorithms, techniques and methodologies we have one this project which will help the peoples who are in the need. So the problem here is that many people goes to hospital so clinic to know how is their health and how much they are improving in the given days, but they have to travel together to know the answers and sometimes the patients may or may not get the results based on various factors such as doctor might be on leave or some whether problem so he might not have come to the hospital and many more reasons will be the rest to avoid all those reasons and confusion we are making a project which will help all those person's and all the patients who are in need to know the condition of their heart disease, and at sometimes if the person has been observing few symptoms and he/she is not sure about the disease he/she is encountered with so this will adapt various diseases in future. so, to avoid that and get to know the heart disease in early stages of the symptoms this disease prediction and classification will help a lotto the various people's ranging from children to teenagers to adults and the senior citizens.

## 1.2 PROJECT PURPOSE

The purpose of making this project called "Face Mask Detection Using Machine Learning Algorithms" is to predict whether a person is wearing a mask or not present in the dataset. Using this information, we can classify and make people to wear mask.

## 1.3 PROJECT FEATURES

The features of Disease Prediction Using Machine Learning are as follows.

- This Project will predict the diseases of the patients based general information using the datasets.
- This is done based on the previous data sets of the hospitals so after comparing it can provide up to 80% of accurate results, and the project is still developing further to get the 100% accurate results.
- With the help of Disease classification, it can classify the disease of the patient and can solve various problems and prevents from various aspects.
- Itprovidessecurityforthesystemsothatnoonecanbreakintothatandnoone can make any changes in the system.
- The disease is classified using the algorithms and the user has to enter the given dataset, in order to get correct accuracy, the user has to enter all the symptoms.
- Here we can easily prepare the data and transform that data into algorithm, which will reduce the overall work of the project.
- To make user more application friendly rather than discussing with others for their disease.
- It provides the necessary options to choose from the types and attributes.

# CHAPTER 2
# LITERATURE SURVEY

## 2.1 MACHINE LEARNING

Machine learning (ML) is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available.

Machine learning is used for the following purposes:

- Prediction — Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups.
- Image recognition — Machine learning can be used for face detection in an image as well. There is a separate category for each person in a database of several people.
- Speech Recognition — It is the translation of spoken words into the text. It is used in voice searches and more. Voice user interfaces include voice dialing, call routing, and appliance control. It can also be used a simple data entry and the preparation of structured documents.
- Medical diagnoses — ML is trained to recognize cancerous tissues.
- Financial industry and trading — companies use ML in fraud investigations and credit checks.

### Types of Machine Learning?

Machine learning can be classified into 3 types of algorithms.

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

### Supervised Learning

In Supervised learning, an AI system is presented with data which is labeled, which means that each data tagged with the correct label. The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

### Types of Supervised learning:

- Classification: A classification problem is when the output variable is a category, such as —red or —blue or —disease and —no disease.

- Regression: A regression problem is when the output variable is a real value, such as —dollars or —weight.

Unsupervised Learning:

In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system's algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.

Types of Unsupervised learning:

- Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Reinforcement Learning

A reinforcement learning algorithm, or agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human by maximizing its reward and minimizing its penalty. It is a type of dynamic programming that trains algorithms using a system of reward and punishment.

**2.1.1 FEATURES OF MACHINE LEARNING**

- It is not Hing but automating the Automation.
- Getting computer stop program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machine learning from data without the help of human so any kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.
- Machine Learning is totally different from traditionally programming,
  Here data and output are given to the computer and in return it gives us the program which provides solution to the various problems. Below is the figure.

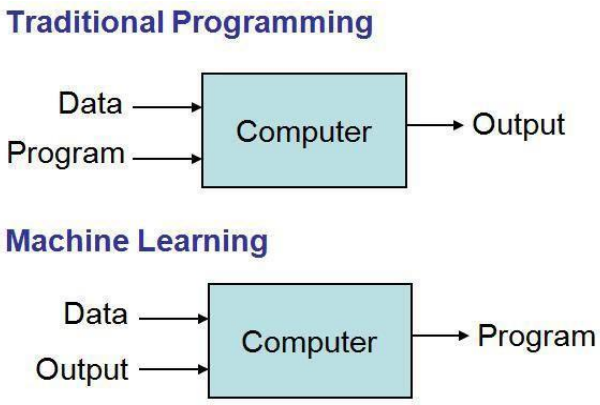



**Fig 2.1.1 Traditional Programming vs Machine Learning**

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There are Many Algorithms in Machine Learning through which we will provide us the exact solution in predicting the disease of the patients.
- How Does Machine Learning Works?
- Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.

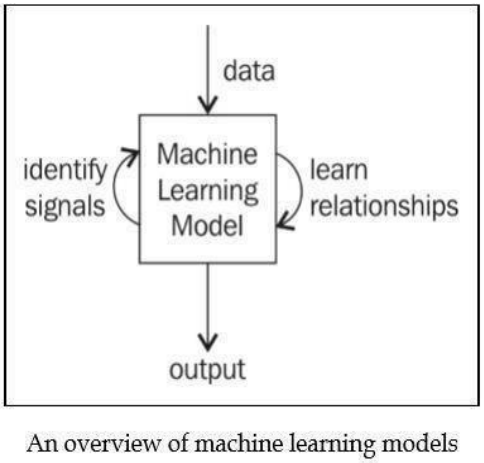



**Fig 2.1.1.2 Machine Learning Model**

- Therearevariousapplicationsinwhichmachinelearningisimplementedsuchas Web search, computing biology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.

## 2.2 EXISTING SYSTEM

The existing system deals with CNN (convolutional neural network) in the face mask detection models, they use clustering, classification, max pooling to train the machine on what is what. The CNN trains the machine with the help of dataset,

around 20% of the images in dataset are used to train the machine and the remaining 80% is used for testing the results. The face mask detection model empathizes with the problems faced by people around the globe due to COVID-19. This system helps in a small way to stop the pandemic from spreading and festering into our lives further. The Person Identification model or the face recognition model as it is popularly called, uses the face recognition library of python to compare images by similarity detection technique. Drawbacks in existing system are the major limitations of existing schemes are as follows: CNN used in existing system are slow and resource hungry, which makes the training process slow. The existing scheme does not detect multiple faces. The existing system does not detect faces from all angles.

## 2.3 PROPOSED SYSTEM

The proposed system develops classification and predictive model that can account for accurate classification grouping and prediction of Face masks on the face of a person. The proposed system will focus on enhancing the prediction by increasing its accuracy and detection probability. This is done by using OpenCV. This system also has the ability to identify the persons who are not wearing the masks and send them a mail notification. The proposed system focuses on how to identify a person wearing a mask on the image/video stream. Help with computer vision and deep learning algorithm by using OpenCV, tensor flow, Keras library. In the proposed architecture, the concept of transfer learning is applied on the backbone to utilize already learned attributes of a powerful pre-trained convolutional neural network in extracting new features for the model.

## 2.4 SOFTWARE DESCRIPTION

### 2.4.1 PYTHON

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is a multi-paradigm programming language. Object-oriented programming and structured-programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and metaobjects. Many other paradigms are supported via extensions, including design by contract and logic

programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector form memory management. It also features dynamic name resolution (late binding), which binds me and variable names during program execution.

Many other paradigms are supported via extensions, including design by contract and logic programming. It is used for:

- web development (server-side),
- software development,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

### 2.4.2 BENEFITS OF PYTHON

- Presence of Third-Party Modules
- Extensive Support Libraries
- Open Source and Community Development
- Learning Ease and Support Available
- User-friendly Data Structures
- Productivity and Speed
- Highly Extensible and Easily Readable Language

## 2.5 DESCRIPTION OF THE PROJECT

This algorithm takes input from the webcam. There are two phases for processing this input data

1.face recognition

2.Feature Extraction

The first stage is facial recognition, which entails detecting a person's face form of a picture. it classifies the image into two categories one is mask if he is wearing a mask and the other one is no mask if he is not wearing a mask.

## 2.6 DESCRIPTION OF THE LIBRARIES

- OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and

its various features we use vector space and perform mathematical operations on these features.

- NumPy

  NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project, and you can use it freely. NumPy stands for Numerical Python. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

- TensorFlow

  TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. Installation of TensorFlow is straightforward if you already have a Python SciPy environment. It can run on single CPU systems, GPUs as well as mobile devices and largescale distributed systems of hundreds of machines.

- Keras

  Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. Keras is relatively straightforward to install if you already have a working Python and SciPy environment. You must also have an installation of Theano or TensorFlow on your system already.

# CHAPTER 3

# REQUIREMENT ANALYSIS AND TECHNICAL DESCRIPTION

## 3.1 REQUIREMENT ANALYSIS

### 3.1.1 FUNCTIONALREQUIREMENTS

AFunctionalrequirementdefinesafunctionofasystemoritscomponent.Afunctionis describe dataset of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behaviouralrequirementsdescribingallcaseswherethesystemusesthefunctionalrequirementsare captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraint son the designer implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

- FunctionalRequirementsconcernswiththespecificfunctionsdeliveredbythesystem.
- So, functional requirements are statements of these vices that the system must provide.
- The function al requirements of the system should be both complete and consistent.
- Completeness means that all the services required by the user should be defined.
- Consistency means that requirements should not have any contradictory definitions.
- The requirements are usually described in an abstract way. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions and soon.
- Take user id and password match it with corresponding file entries. If a match is found, then continue else raise an error message.

### 3.1.2 NON- FUNCTIONAL REQUIREMENTS

- Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.
- The non-functional requirements can be built based on needs of the user, budget constraints, organization policies and etc.

1. Performance requirement: All data entered shall be up to mark and no flaws shall be there for the performance to be 100%.

2. Platform constraints: The main target is to generate an intelligent system to predict the adult height.
3. Accuracy and Precision: Requirements are accuracy and precision of the data.
4. Modifiability: Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort(person-months).
5. Portability: Since mobile phone is handy suites portable and can be carried and used whenever required.
6. Reliability: Requirements about how often the software fails. The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error Prediction, and a strategy for correction.
7. Security: One or more requirements about protection of your system and its data.
8. Usability: Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the cloud can access the cloud to store and retrieve their data with the help of a secret key sent to their email ids. User interface is simple and efficient and easy to use.

MAINTAINABILITY:

In software engineering, maintain ability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using. Net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

SCALABILITY:

Systemiscapableofhandlingincreasetotalthroughputunderanincreasedloadwhen resources (typically hardware) are added. System can work normally under situations such as low band width and large number of users.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the softwarecodebasefeaturetobeabletoreusetheexistingcodeinsteadofcreatingnew code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependent assemblies would have to be configured in such case.

To be used efficiently, all computer software needs certain hardware components or other software resources. These pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time.

- Hardware Requirements
- Software Requirements

### 3.1.3 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Laptop/Desktop

Processor: Intel ®Core™ i3 and above

RAM: 4 GB

### 3.1.4 SOFTWARE REQUIREMENTS

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Operating System: Windows 10

Programming Language: Python

IDE: Python IDLE-3.7

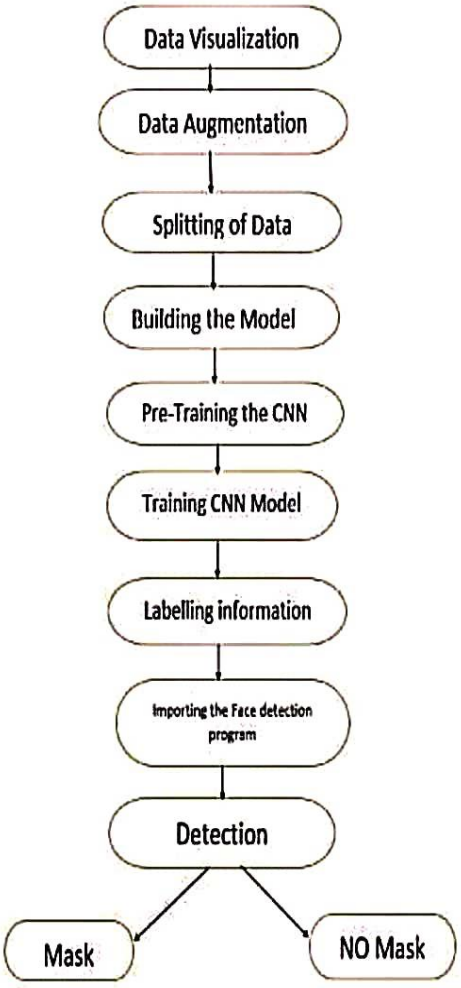
## 3.2 TECHNICAL DESCRIPTION

Convolutional neural network (CNN), a class of artificial neural networks that has become dominant in various computer vision tasks, is attracting interest across a variety of domains, including radiology. CNN is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. This review article offers a perspective on the basic concepts of CNN and its application to various radiological tasks and discusses its challenges and future directions in the field of radiology. Two challenges in applying CNN to radiological tasks, small dataset and overfitting, will also be covered in this article, as well as techniques to minimize them. Being familiar with the concepts and advantages, as well as limitations, of CNN is essential to leverage its potential in diagnostic radiology, with the goal of augmenting the performance of radiologists and improving patient care.

Keras was created to be user friendly, modular, easy to extend, and to work with Python. The API was “designed for human beings, not machines,” and “follows best practices for reducing cognitive load. The aim of image processing is to help the computer to understand the content of an image. OpenCV is a library of programming functions mainly used for image processing. It provides de-facto standard API for computer vision applications. We can solve many real time problems using image processing applications. In this paper, sample real time image processing applications of OpenCV are discussed along with steps.

# CHAPTER 4

## 4.1 SYSTEM ARCHITECTURE




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

## 4.2 DATA FLOW DIAGRAM

The data flow diagram of the project face mask detection using machine learning consist of all the various aspects a normal flow diagram requires. This data flow diagram show from starting the model flows from one-step to another, like enter the dataset into the system then enters all the information's and all other general information that goes into the system, compares with the classification model and if true is provides the appropriate results otherwise it shows the details where the information is wrong.

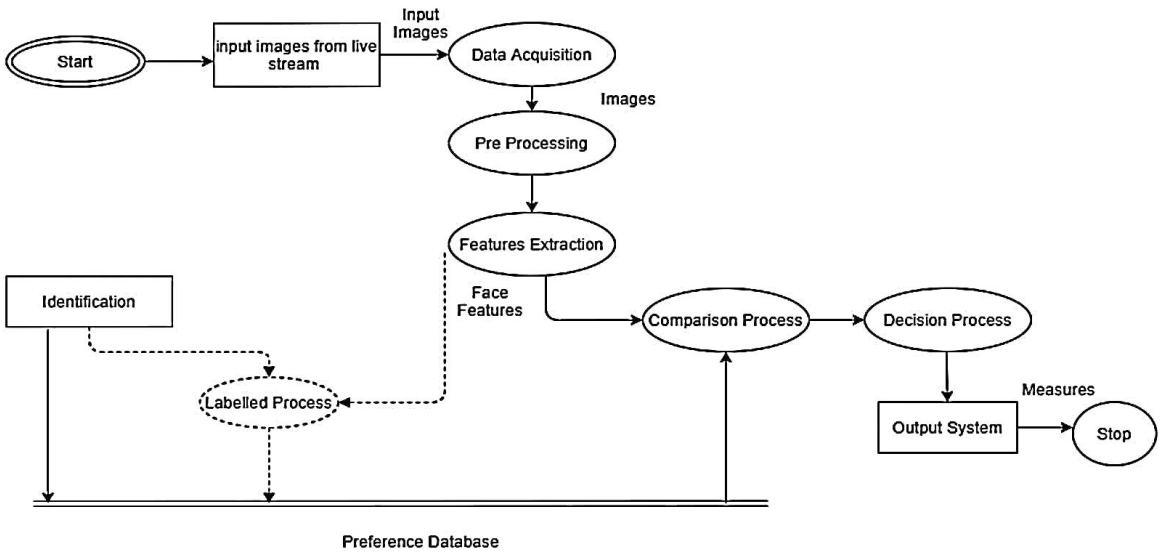



Fig 4.2 Data Flow Diagram

## 4.3 CLASS DIAGRAM

Face Mask Detection using machine learning consist of class diagram that all the other application that consists of the basic class diagram, here the class diagram is the basic entity that is required in order to carry on with the project. Class diagrams consist of information about all the classes that is used and all the related datasets, and all the other necessary attributes and their relationships with other entities, all these information is necessary in order to use the concept of the classification, where the dataset consists of all necessary information.

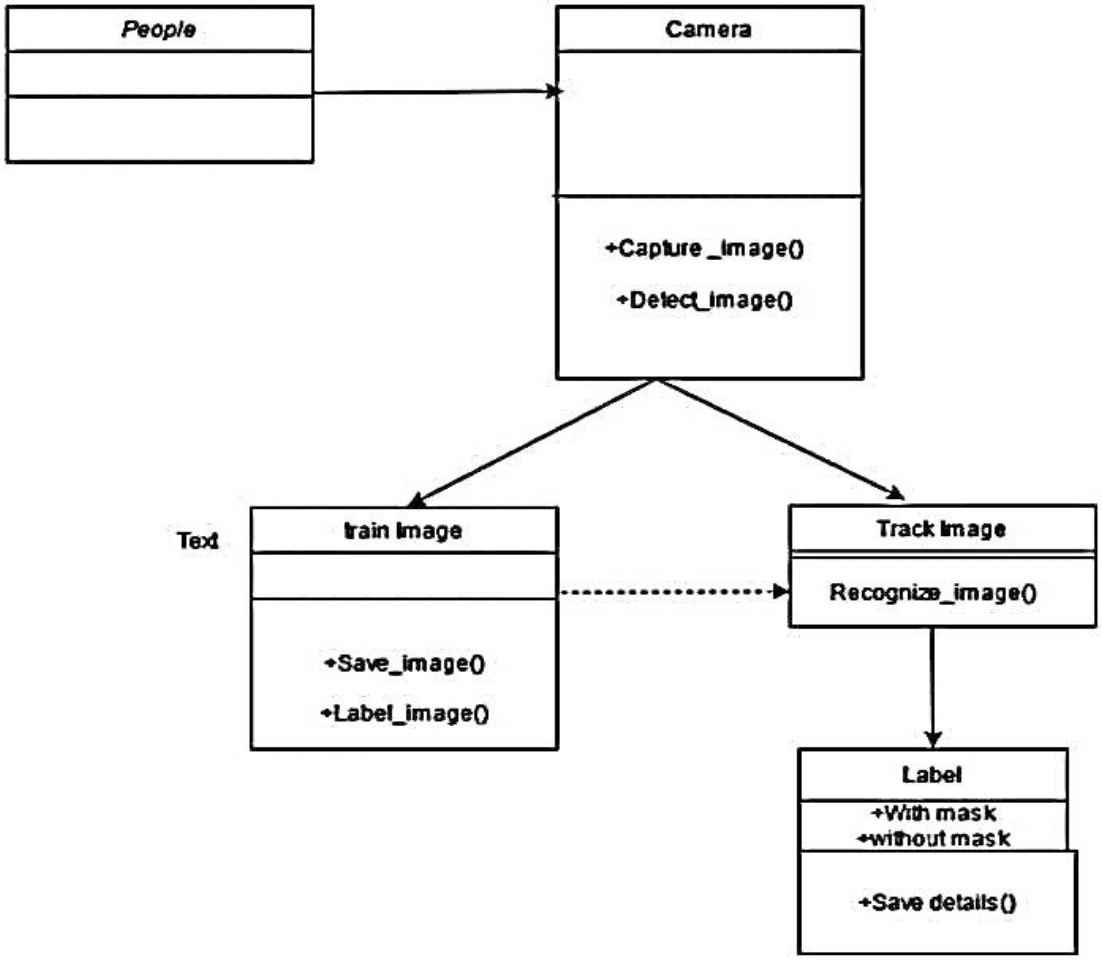



Fig 4.3 Class Diagram

## 4.4 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development. Here component diagram consists of all major components that is used to build a system. So, Design, Algorithm, File System and Datasets all are linked to one another. Datasets are used to compare the results and algorithm is used to process those results and give a correct accuracy and design UI is used to show the result in an appropriate way in the system and file system is used to store the user data. So, like this all components are inter linked to each other.

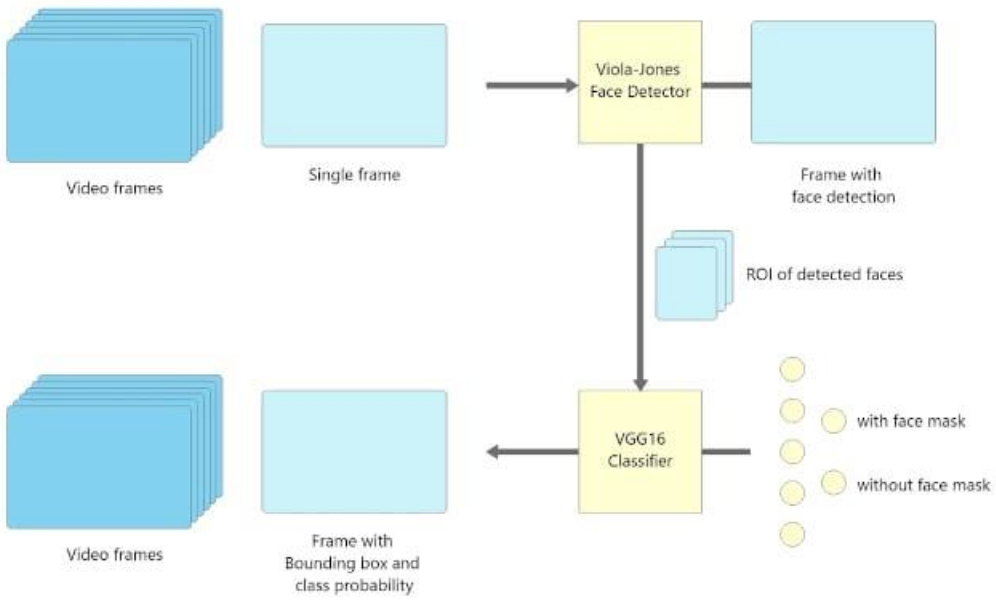



Fig 4.4Component Diagram

## 4.5 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

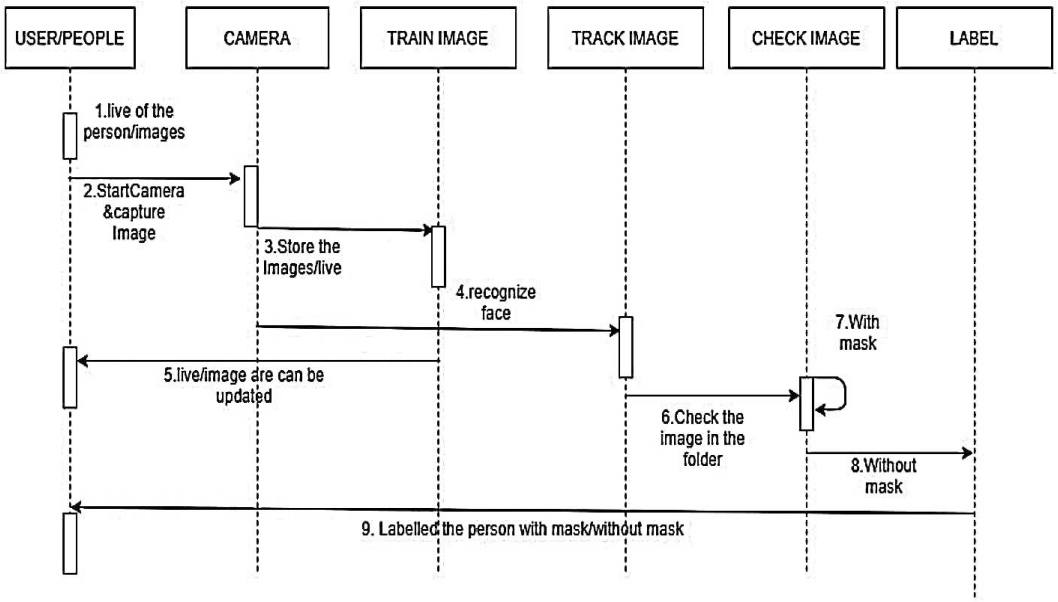



Fig. 4.5Sequence Diagram

# CHAPTER 5
# IMPLEMENTATION

## 5.1 CNN

```
# Import necessary packages
import os
import cv2
import time
from tqdm import tqdm
import numpy as np
import tensorflow as tf
from sklearn. model selection import train_test_split
from tensorflow. keras.models import Sequential, load model
from tensorflow.keras.layers import Conv2D, Dense, MaxPooling2D, Fl
atten, Activation, Dropout


img_size = 100
data Dir = r'Data'    # root data directiory
CATEGORIES = os. listdir(datadir)
print (CATEGORIES)


# Define two empty list to contain image data
x, y = [], []

def PreProcess():
   for category in CATEGORIES:
      path = os. path.join(datadir, category)
      class Index = CATEGORIES. Index(category)
      print(path)
      for imgs in tqdm (os. listdir(path)):
         img_arr = cv2.imread(os. path.join(path, imgs))
```

```
            # resize the image
            resized_array = cv2.resize(img_arr, (img_size, img_size))
            cv2.imshow("images", resized_array)
            cv2.waitKey(1)
            resized_array = resized_array/255.0
            x.append(resized_array)
            y.append(classIndex)

PreProcess()
cv2.destroyAllWindows()


# Split data for training and testing
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20, ra
ndom_state=42)



# Convert and resize the data to a numpy array
X_train = np.array(X_train).reshape(-1, img_size, img_size, 3)
y_train = np.array(y_train)
X_test = np.array(X_test).reshape(-1, img_size, img_size, 3)
y_test = np.array(y_test)



batch size = 32
epochs = 15


# Create the model architecture
```

```
model = Sequential()

model.add(Conv2D(64,(3, 3), input_shape=(img_size, img_size, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dense(16, activation='relu'))


model.add(Dense(len(CATEGORIES)))
model.add(Activation('softmax'))

# compile the model

model.compile(optimizer='adam', loss = 'sparse_categorical_crossentr
opy', metrics=['accuracy'])
model.summary()
```

```
t1 = time.time()

# fit the model
model.fit(X_train, y_train, batch_size = batch_size, epochs=epochs, val
idation_split=0.3, verbose = 1)
model.save('{}.h5'.format("model2"))

t2 = time.time()
print('Time taken: ',t2-t1)


print("Model evaluation : ")
validation_loss, validation_accuracy = model.evaluate(X_test, y_test)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 98, 98, 64)        1792

activation (Activation)      (None, 98, 98, 64)        0

max_pooling2d (MaxPooling2D  (None, 49, 49, 64)        0
)

conv2d_1 (Conv2D)            (None, 47, 47, 256)       147712

activation_1 (Activation)    (None, 47, 47, 256)       0

max_pooling2d_1 (MaxPooling  (None, 23, 23, 256)       0
2D)

conv2d_2 (Conv2D)            (None, 21, 21, 128)       295040

activation_2 (Activation)    (None, 21, 21, 128)       0

dropout (Dropout)            (None, 21, 21, 128)       0

conv2d_3 (Conv2D)            (None, 19, 19, 32)        36896

activation_3 (Activation)    (None, 19, 19, 32)        0
```

max_pooling2d_2 (MaxPooling  (None, 9, 9, 32)  0
2D)

dropout_1 (Dropout)  (None, 9, 9, 32)  0

flatten (Flatten)  (None, 2592)  0

dense (Dense)  (None, 100)  259300

dense_1 (Dense)  (None, 16)  1616

dense_2 (Dense)  (None, 2)  34

activation_4 (Activation)  (None, 2)  0

=================================================================
Total params: 742,390
Trainable params: 742,390
Non-trainable params: 0
_____

## TESTNG

```
# Detection function

def get detection(frame):


    height, width, channel = frame.shape


    # Convert frame BGR to RGB colorspace


    imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Detect results from the frame


    result = face_detection.process(imgRGB)
```

```
    try:
        for count, detection in enumerate(result.detections):

            # Print(detection)



            # Extract bounding box information

            box = detection. location_data.relative_bounding_box

            x, y, w, h = int(box.xmin*width), int(box.ymin * height),
int(box.width*width), int(box.height*height)

    # If detection is not available then pass
    except:
        pass

    return x, y, w, h

CATEGORIES = ['no_mask', 'mask']
```

```
cap = cv2.VideoCapture(0)
c=0
while True:
    _, frame = cap.read()
    img = frame.copy()
    try:
        x, y, w, h = get_detection(frame)

        crop_img = img[y:y+h, x:x+w]

        crop_img = cv2.resize(crop_img, (100, 100))

        crop_img = np.expand_dims(crop_img, axis=0)

        # get the prediction from the model.
        prediction = model.predict(crop_img)
        print(prediction)
        index = np.argmax(prediction)
        res = CATEGORIES[index]
        print(res)
        if(res=='no_mask'):
            c=c+1
```

```
        else:
            c=0
        if index == 0:
            color = (0, 0, 255)
        else:
            color = (0, 255, 0)
        cv2.rectangle(frame, (x, y), (x+w, y+h), color, 2)
        cv2.putText(frame, res, (x, y-10),
cv2.FONT_HERSHEY_SIMPLEX,
                            0.8, color, 2, cv2.LINE_AA)
        if c>50:
            os.startfile("F:\\m1.ogg")
            c=0

    except:
        pass

    cv2.imshow("frame", frame)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
```

cv2.destroyAllWindows()

Time taken:  333.8416941165924

Model evaluation :

1/9 [==>.....] - ETA: 1s - loss: 0.0731 - accuracy: 0.9375
2/9 [=====>.....] - ETA: 1s - loss: 0.0404 - accuracy: 0.9688
3/9 [=========>.....] - ETA: 1s - loss: 0.0701 - accuracy: 0.9583
4/9 [============>.....] - ETA: 1s - loss: 0.0548 - accuracy: 0.9688
5/9 [===============>.....] - ETA: 0s - loss: 0.0737 - accuracy: 0.9625
6/9 [==================>.....] - ETA: 0s - loss: 0.0996 - accuracy: 0.9635
7/9 [======================>.....] - ETA: 0s - loss: 0.0882 - accuracy: 0.9688
8/9 [==========================>....] - ETA: 0s - loss: 0.0783 - accuracy: 0.9727
9/9 [==============================] - ETA: 0s - loss: 0.0763 - accuracy: 0.9734
9/9 [==============================] - 2s 240ms/step - loss: 0.0763 - accuracy: 0.9734

# CHAPTER 6

# TESTING

## TYPES OF TESTS

### UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

### INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### VALIDATION TESTING

An Engineering Validation Test (EVT) is performed on first Engineering prototypes, to ensure that the basic unit is performs to design goals and specifications. It is important in identifying design problems and solving the as early in the design cycle as possible, is the key to keeping projects on time and within budget. Verification is a Quality control process that to evaluate whether a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process. Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

## SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integratedwithanyapplicablehardwaresystem.Systemtestingisamorelimited typeoftesting;itseekstodetectdefectsbothwithinthe"inter-assemblages"and also within the system as a whole .System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) or System Requirement Specification(SRS).

## The testing process overview is as follows:

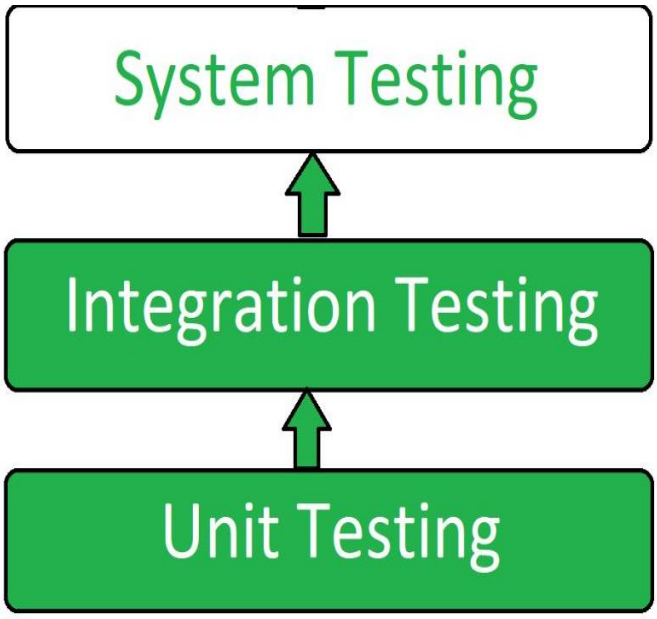



**Fig 6.1 The Testing Process**

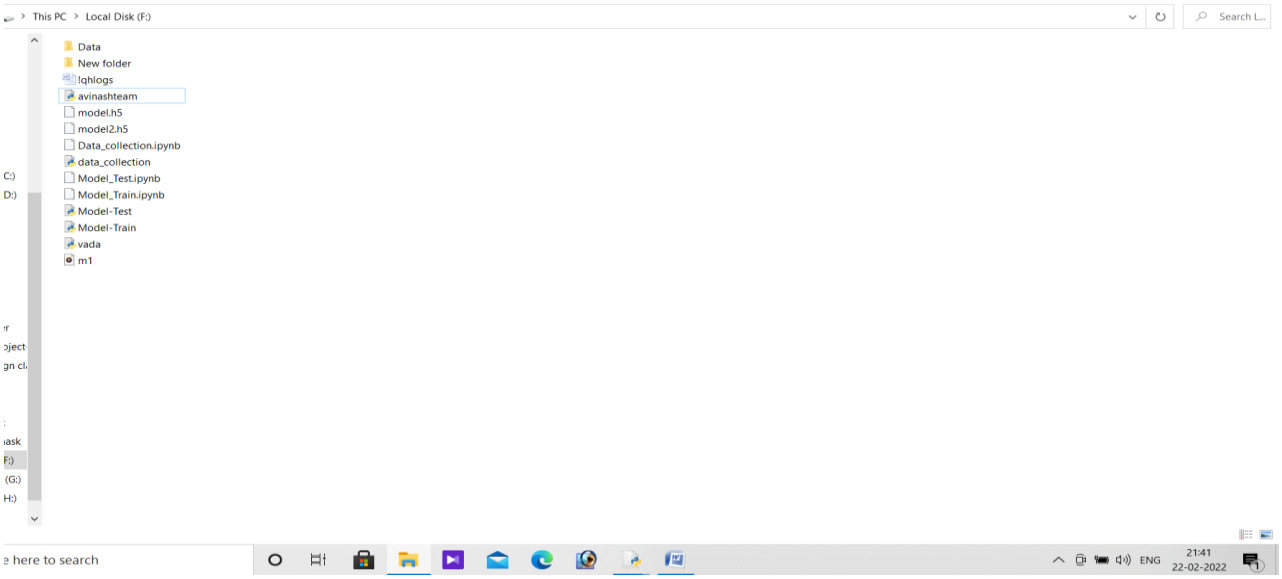
| | |
|---|---|
| Serial Number of Test Case | 1132 |
| Module Under Test | Classification Result |
| Description | Dataset contains the images of people with and without masks |
| Input | Dataset |
| Output | Based on information in the dataset to calculate the accuracy which will be having high and then classify. |
| Remarks | Test Successful. |

**Table 6.1 Test Case for Prediction Result**

# CHAPTER 7

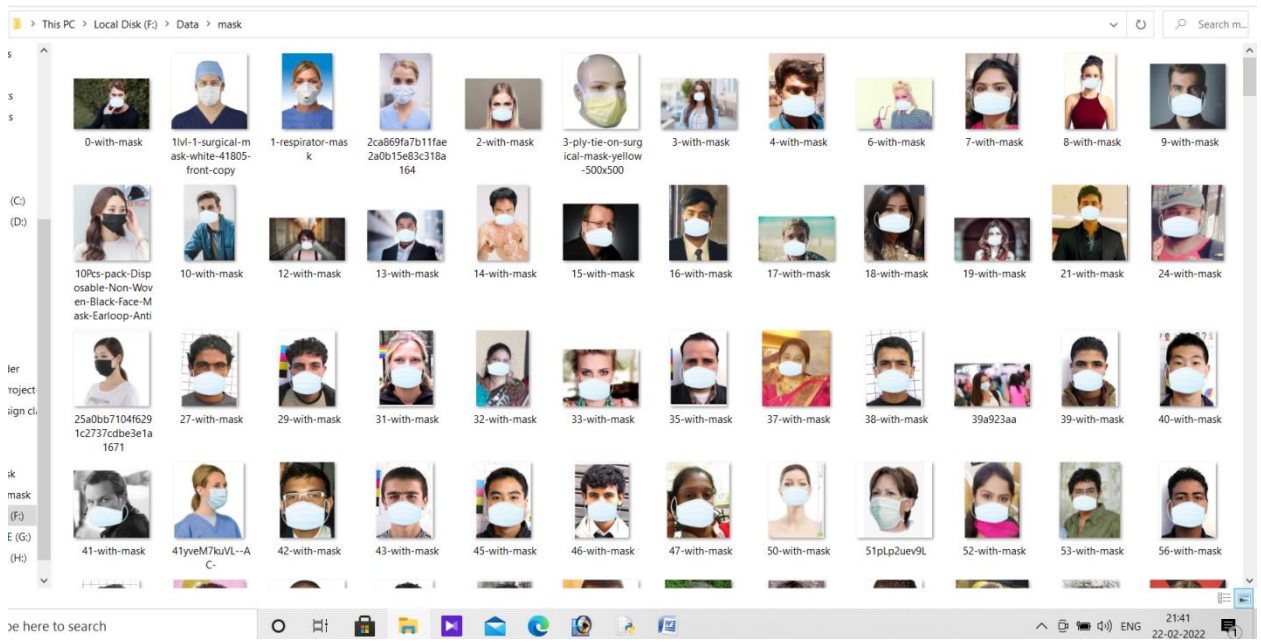
# SCREENSHOTS

## File location:

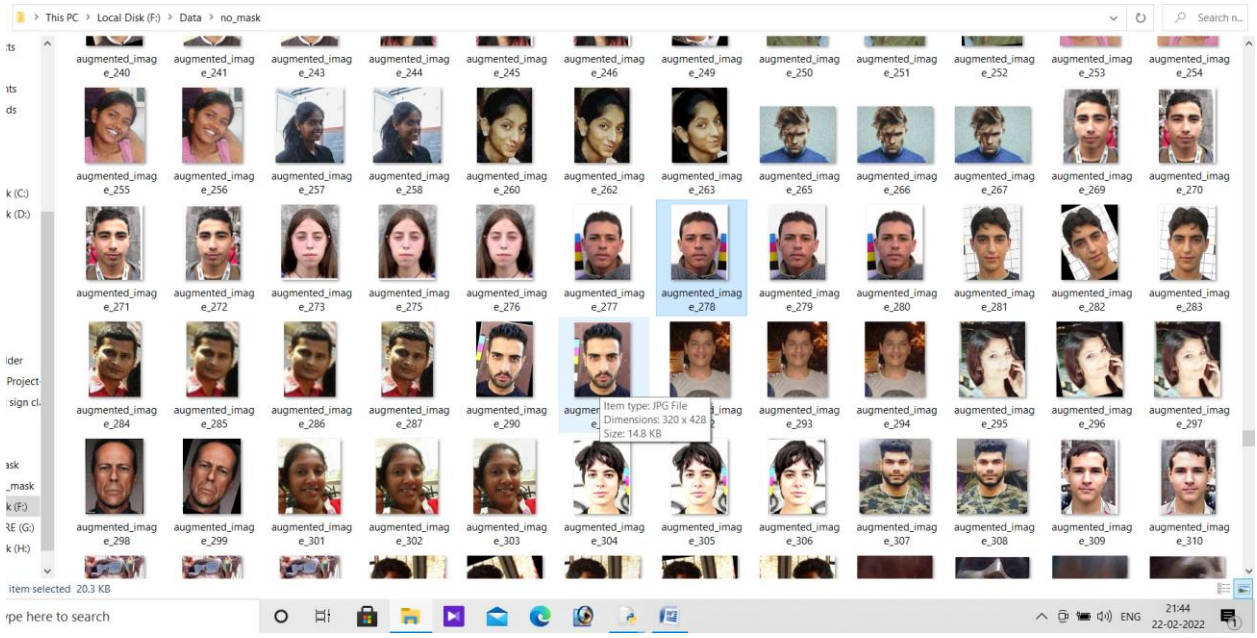



## Datasets:

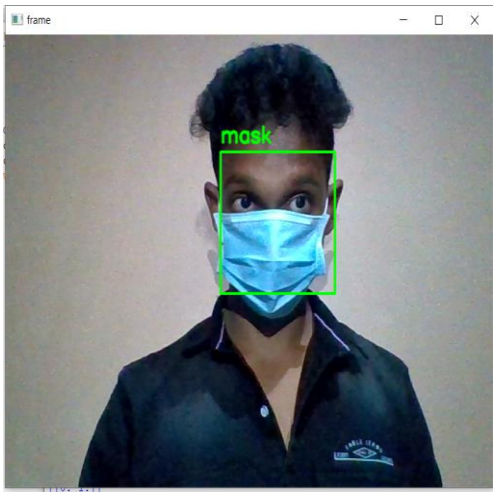
## With mask:

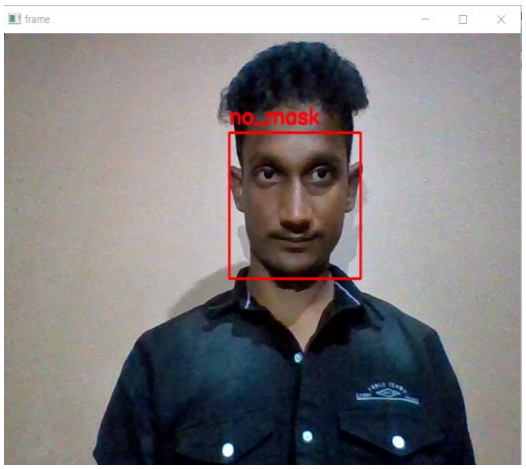


## Without mask:




## Mask recognition:



# CHAPTER 8

# CONCLUSION AND FUTURE ENHANCEMENT

## 8.1 CONCLUSION

To moderate the spread of the covid-19 pandemic, measures should be taken. We have demonstrated facemask detector using convolutional neural network. To train, validate and test the model, we Utilized the dataset that consist of masked faces pictures and exposed faces pictures. The model was in duced on pictures and live video transfers. To choose a base model, we assessed the measurements Like precision, accuracy, and recall and chose OpenCV architecture. It is additionally computationally Efficient using mobilenetv2 which makes it simpler to introduce the model to inserted frameworks. This face mask detector can be sent in numerous regions like shopping centres, air terminals and Other substantial traffic places to screen people in general and to dodge the spread of the infection by Checking who is following essential rules and who isn't.

## 8.2 FUTURE ENHANCEMENT

- Facility for modifying user detail
- More interactive user interface.
- Facilities for Backup creation.
- Can be done as Webpage.
- Can be done as Mobile Application.
- More Details and Latest Diseases
- Nowadays we see automated doors in many places like restaurants, shopping malls, and offices.
- These automated doors were operated based on data coming from camera and sensors.
- If a person is passes to the door, it will be automatically opened.
- Based on the present scenario there is a lot of necessity to wear mask.
- The output of this project can be used for operating the automated doors.
- The door will be opened only when the person is wearing the mask. If he is not wearing the mask, then advise him to wear the mask by means of voice message

# CHAPTER 9

# REFERENCES


- [1] Z. Allam and D. S. Jones, “On the Coronavirus (COVID-19) Outbreak and the Smart City Network: Universal Data Sharing Standards Coupled with Artificial Intelligence (AI) to Benefit Urban Health Monitoring and Management,” Healthcare, vol. 8, no. 1, p. 46, 2020.
- [2] X. Wang, X. Le and Q. Lu, "Analysis of China’s Smart City Upgrade and Smart Logistics Development under the COVID-19 Epidemic", J. Phys. Conf. Ser., vol. 1570, pp. 012066, 2020. 8. Karthick, R.
- [3] Karthick, R., et al. “A Geographical Review: Novel Coronavirus ( COVID-19) Pandemic.” A Geographical Review: Novel Coronavirus ( COVID-19) Pandemic (October 16, 2020). Asian Journal of Applied Science and Technology (AJAST) (Quarterly International Journal) Volume 4 (2020): 44-50.
- [4] R. Jaiswal, A. Agarwal and R. NEGI, "Smart Solution for Reducing the COVID-19 Risk using Smart City Technology", IET Smart Cities, vol. 2, pp. 82-88, 2020.
- [5] Karthick, R., et al. “Overcome the challenges in bio-medical instruments using IOT–A review.” Materials Today: Proceedings (2020). https://doi.org/10.1016/j.matpr.2020.08.420.