



A project report on

## **AI FACE MASK DETECTION**

Submitted in partial fulfillment of the requirements for the Degree of

B. Tech in Computer Science of Engineering

by

**Arjun Tanpure (1829203)**

**Joyjit Nath (1829156)**

under the guidance of

**Dr. Dipti Dash**

School of Computer Engineering  
Kalinga Institute of Industrial Technology

Deemed to be University  
Bhubaneswar

March 2021



## CERTIFICATE

This is to certify that the project report entitled "**AI Face Mask Detection**" submitted by

**ARJUN TANPURE**                   **1829203**

**JOYJIT NATH**                   **1829156**

in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology in Discipline of Engineering** is a bonafide record of the work carried out under my(our) guidance and supervision at School of Computer Science Engineering, Kalinga Institute of Industrial Technology, Deemed to be University.

Signature of Supervisor 2 (if applicable)

Signature of Supervisor 1

NAME OF THE SUPERVISOR 2

NAME OF THE SUPERVISOR 1

Academic affiliation

Academic affiliation

Organization

Organization

---

**The Project was evaluated by us on DD/MM/YYYY**

EXAMINER 1

EXAMINER 2

EXAMINER 3

EXAMINER 4

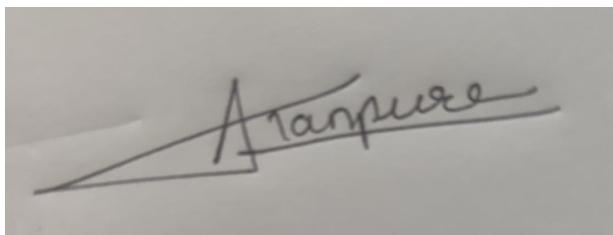
## ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my Professor **Dr. Dipti Dash**, who has the attitude and the substance of a genius: she continually and convincingly conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without her guidance and persistent help this dissertation would not have been possible.

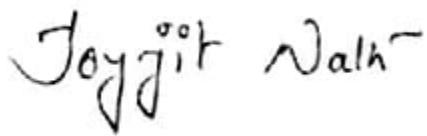
I would also like to thank my research supervisors, for their advice to finish the project with ease and on time and also shared their support, either morally which also helped us in doing a lot of Research and we came to know about so many new things.

Without these people this project wouldn't be successful or possible. That's why we should like to say, "thank you.

**Arjun Tanpure**



**Joyjit Nath**



## ABSTRACT

The dataset is imported to the Jupyter notebook with the help of library. Initial stage is to divide the dataset into two categories “0 and 1” and label them ‘without\_mask’ and ‘with\_mask’. Next step is to train our model with the help of tensorflow-keras and Scikit-learn. We will be taking 20 epochs in our checkpoint.

Now it's time to load our model with keras load\_model library, while loading the model labels are 0 for ‘NO MASK’ and 1 for ‘MASK’ and colours given are Red (0,0,255) and Green (0,255,0). While loading our model for our model to predict correctly we have used the ‘haarcascade\_frontalface\_default.xml’ file.

Haarcascade\_frontalface\_default.xml file is the file. Then we give rectangle specifications and finally after running the last cell we get our live result in front of the camera whether a person is wearing a mask or not.

## TABLE OF CONTENTS

Abstract	: 4
Table of Contents	: 5
List of Figures	: 6
CHAPTER 1: INTRODUCTION	: 7
1.1 Techs and overview of the project	: 7
1.1.1 About and Aim of the project.	: 7
1.2 Purpose of the Mask detection Software	: 7
CHAPTER 2: BACKGROUND/BASIC CONCEPTS	: 8
2.1 Theory of Mask Detection	: 8
2.1.1 Literature Review	: 8
2.1.2 Advantages of the Model	: 8
CHAPTER3: PROJECT ANALYSIS/ PROJECT IMPLEMENTATION	: 9
3.1 Project	: 9
CHAPTER 4: RESULTS AND DISCUSSION	: 1
	4
CHAPTER 5: CONCLUSION & FURTHER WORK	: 1
	5
REFERENCES	: 1
	6

## LIST OF FIGURES

<b>Figure ID</b>	<b>Figure Title</b>	<b>Page</b>
1.1	Techs and overview of the project	7
1.2	Purpose of the Mask detection Software	7
2.1	Theory of Mask Detection	8
3.1	Project	9
4.1	Analysis of Result	14
5.1	Conclusion	15
5.2	Future Work	25

## CHAPTER 1

### INTRODUCTION

#### 1.1 Techs and Overview of the Project –

##### 1.1.1 About the project –

---

The year 2020 has shown mankind some mind-boggling series of events amongst which the COVID-19 pandemic is the most life-changing event which has startled the world since the year began. Affecting the health and lives of masses, COVID-19 has called for strict measures to be followed in order to prevent the spread of disease. From the very basic hygiene standards to the treatments in the hospitals, people are doing all they can for their own and the society's safety; **face masks** are one of the personal protective equipment. People wear face masks once they step out of their homes and authorities strictly ensure that people are wearing face masks while they are in groups and public places. To monitor that people are following this basic safety principle, a strategy should be developed. A face mask detector system can be implemented to check this. Face mask detection means to identify whether a person is wearing a mask or not.

---

**Aim of the project** - ‘AI mask Detection’ is our project name and it comes under Artificial intelligence and the tech used is python and all the necessary libraries. To develop an Artificial intelligence model which can Detect whether a person is wearing a mask or not.

#### 1.2 Purpose of the Mask detection Software -

In this Pandemic, as some of the people are not following the government guidelines and violating the rules by not wearing masks. So, our software will help government officials to detect whether a person is wearing a mask or not in crowded areas where it is needed and take the required action against them.

---

## CHAPTER 2

### BACKGROUND

---

#### **2.1 Theory of Mask Detection –**

Mask Detection is the process in which cameras are installed on the public gathering places by the government, and the technical hub is installed nearby, in our process the camera captures the video of people trespassing in the public places as soon as the software detects any-one without mask software makes alert sign and the photo of the person is captured by the officials and the required actions are taken-out.

##### **2.1.1 Literature Review –**

Object detection is one of the trending topics in the field of image processing and computer vision. Ranging from small scale personal applications to large scale industrial applications, object detection and recognition is employed in a wide range of industries. Some examples include image retrieval, security and intelligence, OCR, medical imaging and agricultural monitoring. In object detection, an image is read and one or more objects in that image are categorized. The location of those objects is also specified by a boundary called the bounding box. Traditionally, researchers used pattern recognition to predict faces based on prior face models. A breakthrough face detection technology then was developed named as **Viola Jones detector** that was an optimized technique of using **Haar**, digital image features used in object recognition. However, it failed because it did not perform well on faces in dark areas and non-frontal faces. Since then, researchers are eager to develop new algorithms based on deep learning to improve the models. There are various methods of object detection based on deep learning which are divided into two categories: one stage and two stage object detectors.

##### **2.1.2 Advantages of the model –**

Our model helps government officials to reduce covid affect in public areas. Our model is also responsible and useful in places like schools and hospitals. Government can generate some revenue from the fine imposed on people who are violating safety rules as in like Metro, Airports, Offices and Educational Institutions etc.

## CHAPTER 3

### PROJECT ANALYSIS/ PROJECT IMPLEMENTATION

#### 3.1 Project –

Folder Structure –

📁	.ipynb_checkpoints	22-04-2021 18:47	File folder
📁	dataset	22-04-2021 18:12	File folder
📁	model-001.model	24-04-2021 16:49	File folder
📁	model-002.model	24-04-2021 16:50	File folder
📁	model-003.model	24-04-2021 16:51	File folder
📁	model-004.model	24-04-2021 12:05	File folder
📁	model-005.model	24-04-2021 16:52	File folder
📁	model-006.model	24-04-2021 16:53	File folder
📁	model-007.model	24-04-2021 16:54	File folder
📁	model-008.model	24-04-2021 12:08	File folder
📁	model-009.model	24-04-2021 16:55	File folder
📁	model-010.model	24-04-2021 12:10	File folder
📁	model-015.model	24-04-2021 17:00	File folder
📁	model-017.model	26-04-2021 12:30	File folder
📄	data.npy	24-04-2021 16:49	NPY File 1,07,501 KB
📝	GUI_Screenshots.pdf	26-04-2021 12:36	Adobe Acrobat D... 2,397 KB
📄	haarcascade_frontalface_default.xml	20-04-2021 13:25	XML Document 1,226 KB
💻	mask_detection 6th sem minor project...	24-04-2021 17:26	IPYNB File 15 KB
💻	mask_detection_GUI.py	26-04-2021 12:46	Python File 4 KB
📝	Minor Project Report.pdf	26-04-2021 12:25	Adobe Acrobat D... 129 KB
📄	target.npy	24-04-2021 16:49	NPY File 11 KB

This project is built on Jupyter Notebook; it is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the I-Python project, which used to have an I-Python Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the I-Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

- In the beginning, we imported libraries like keras, OpenCV, NumPy, and OS.

The screenshot shows a Jupyter Notebook interface with several tabs at the top: 'Machine Learning, Data Science', 'Desktop', 'mask\_detection 6th sem minor project' (active), and 'T&T lab 1 assignment by Arjun'. Below the tabs is a toolbar with icons for file operations and cell execution. The main area contains code cells:

```
In [11]: pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\kiit\anaconda3\envs\new_env\lib\site-packages (4.5.1.48)
Requirement already satisfied: numpy>=1.17.3 in c:\users\kiit\anaconda3\envs\new_env\lib\site-packages (from opencv-python) (1.19.2)

In [1]: import cv2

In [2]: import numpy as np

In [3]: from keras.utils import np_utils

In [4]: import os

In [5]: data_path='C:/Users/KIIT/Desktop/dataset'

In [6]: categories=os.listdir(data_path)
labels=[i for i in range(len(categories))]
label_dict=dict(zip(categories,labels))
print(label_dict)
print(categories)
print(labels)

{'without_mask': 0, 'with_mask': 1}
['without mask', 'with mask']
```

- Here we categorize our data into two different formats, one is if we have a mask then '1' and without mask '0'.

The screenshot shows a Jupyter Notebook interface with several tabs at the top: 'Machine Learning, Data Science', 'Desktop', 'mask\_detection 6th sem minor project' (active), and 'T&T lab 1 assignment by Arjun'. Below the tabs is a toolbar with icons for file operations and cell execution. The main area contains code cells:

```
In [6]: categories=os.listdir(data_path)
labels=[i for i in range(len(categories))]
label_dict=dict(zip(categories,labels))
print(label_dict)
print(categories)
print(labels)

{'without_mask': 0, 'with_mask': 1}
['without mask', 'with mask']
[0, 1]

In [7]: img_size=100
data=[]
target=[]

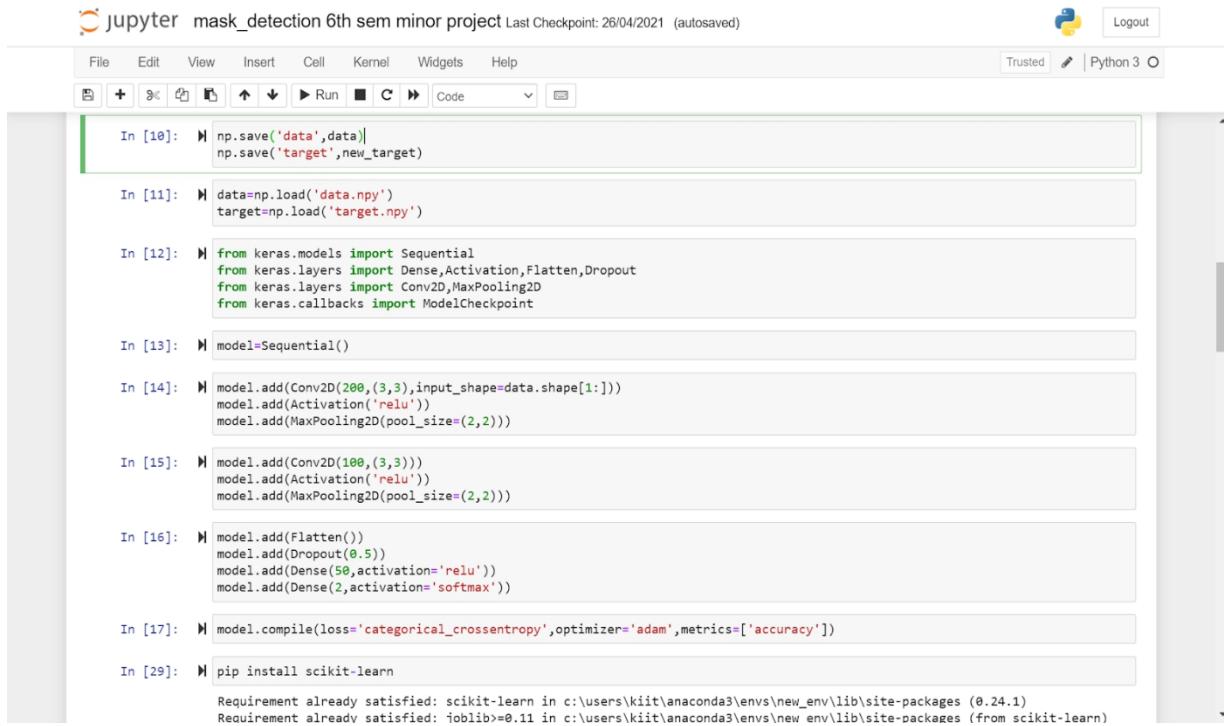
In [8]: for category in categories:
    folder_path=os.path.join(data_path,category)
    img_names=os.listdir(folder_path)

    for img_name in img_names:
        img_path=os.path.join(folder_path,img_name)
        img=cv2.imread(img_path)
        gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        resized=cv2.resize(gray,(img_size,img_size))
        data.append(resized)
        target.append(label_dict[category])

In [9]: data=np.array(data)/255.0
data=np.reshape(data,(data.shape[0],img_size,img_size,1))
target=np.array(target)
new_target=np_utils.to_categorical(target)

In [10]: np.save('data',data)
np.save('target',new_target)
```

- then we use NumPy, we give data and target to our model, then we use sequential, we take our activation function as ‘relu’
- After that we compile our model, our optimizer is ‘adam’. the with the help of scikit learn we move further.



```

jupyter mask_detection 6th sem minor project Last Checkpoint: 26/04/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 Logout

In [10]: np.save('data',data)
np.save('target',new_target)

In [11]: data=np.load('data.npy')
target=np.load('target.npy')

In [12]: from keras.models import Sequential
from keras.layers import Dense,Activation,Flatten,Dropout
from keras.layers import Conv2D,MaxPooling2D
from keras.callbacks import ModelCheckpoint

In [13]: model=Sequential()

In [14]: model.add(Conv2D(200,(3,3),input_shape=data.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

In [15]: model.add(Conv2D(100,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

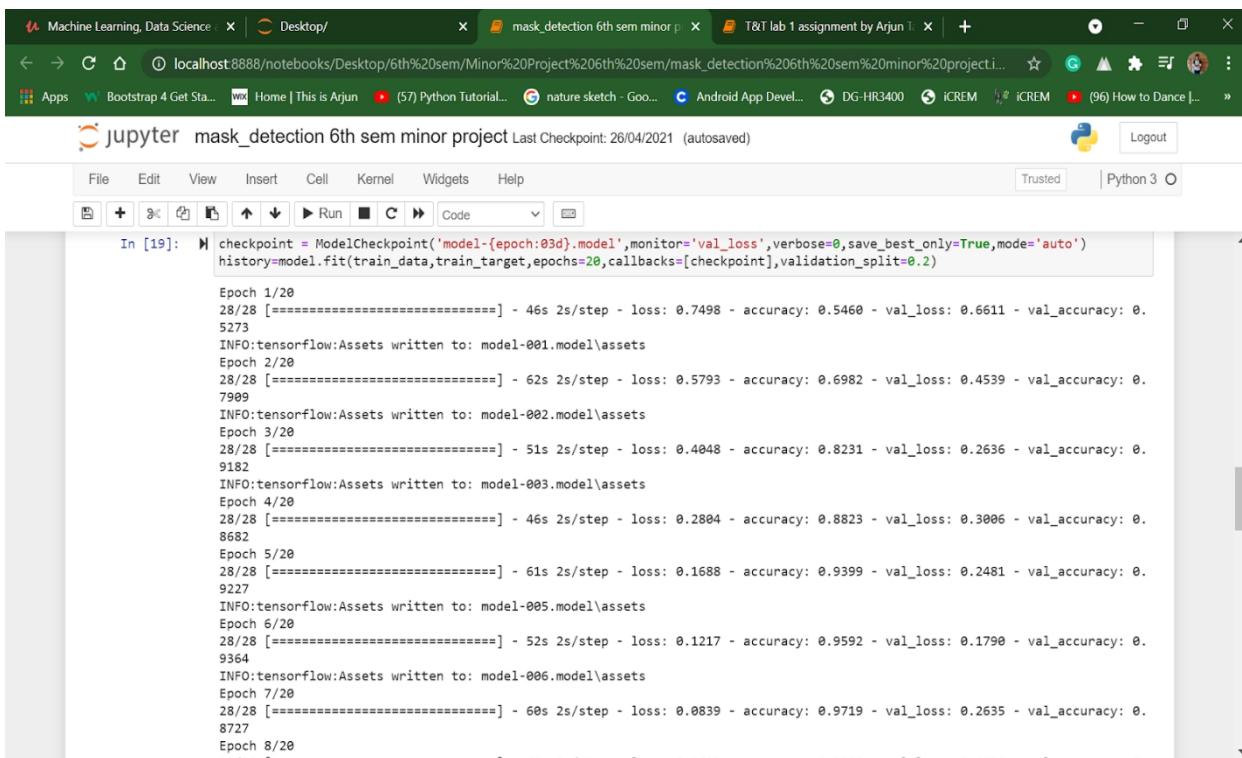
In [16]: model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(50,activation='relu'))
model.add(Dense(2,activation='softmax'))

In [17]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

In [29]: pip install scikit-learn
Requirement already satisfied: scikit-learn in c:\users\kiit\anaconda3\envs\new_env\lib\site-packages (0.24.1)
Requirement already satisfied: joblib>=0.11 in c:\users\kiit\anaconda3\envs\new_env\lib\site-packages (from scikit-learn)

```

- Now we will fit our model, the number of epochs taken are 20.



```

Machine Learning, Data Science | Desktop/ mask_detection 6th sem minor project | T&T lab 1 assignment by Arjun Ti... | +
localhost:8888/notebooks/Desktop/6th%20sem/Minor%20Project%206th%20sem/mask_detection%206th%20sem%20minor%20project...
Apps Bootstrap 4 Get Sta... Home | This is Arjun (57) Python Tutorial... nature sketch - Goo... Android App Devel... DG-HR3400 ICREM ICREM (96) How to Dance [...]
jupyter mask_detection 6th sem minor project Last Checkpoint: 26/04/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [19]: checkpoint = ModelCheckpoint('model-{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')
history=model.fit(train_data,train_target,epochs=20,callbacks=[checkpoint],validation_split=0.2)

Epoch 1/20
28/28 [=====] - 46s 2s/step - loss: 0.7498 - accuracy: 0.5460 - val_loss: 0.6611 - val_accuracy: 0.5273
INFO:tensorflow:Assets written to: model-001.model\assets
Epoch 2/20
28/28 [=====] - 62s 2s/step - loss: 0.5793 - accuracy: 0.6982 - val_loss: 0.4539 - val_accuracy: 0.7909
INFO:tensorflow:Assets written to: model-002.model\assets
Epoch 3/20
28/28 [=====] - 51s 2s/step - loss: 0.4048 - accuracy: 0.8231 - val_loss: 0.2636 - val_accuracy: 0.9182
INFO:tensorflow:Assets written to: model-003.model\assets
Epoch 4/20
28/28 [=====] - 46s 2s/step - loss: 0.2804 - accuracy: 0.8823 - val_loss: 0.3006 - val_accuracy: 0.8682
Epoch 5/20
28/28 [=====] - 61s 2s/step - loss: 0.1688 - accuracy: 0.9399 - val_loss: 0.2481 - val_accuracy: 0.9227
INFO:tensorflow:Assets written to: model-005.model\assets
Epoch 6/20
28/28 [=====] - 52s 2s/step - loss: 0.1217 - accuracy: 0.9592 - val_loss: 0.1790 - val_accuracy: 0.9364
INFO:tensorflow:Assets written to: model-006.model\assets
Epoch 7/20
28/28 [=====] - 60s 2s/step - loss: 0.0839 - accuracy: 0.9719 - val_loss: 0.2635 - val_accuracy: 0.8727
Epoch 8/20

```

```

INFO:tensorflow:Assets written to: model-008.model\assets
Epoch 9/20
28/28 [=====] - 52s 2s/step - loss: 0.0742 - accuracy: 0.9690 - val_loss: 0.1360 - val_accuracy: 0.9500
INFO:tensorflow:Assets written to: model-009.model\assets
Epoch 10/20
28/28 [=====] - 47s 2s/step - loss: 0.0600 - accuracy: 0.9781 - val_loss: 0.1456 - val_accuracy: 0.9455
Epoch 11/20
28/28 [=====] - 48s 2s/step - loss: 0.0592 - accuracy: 0.9815 - val_loss: 0.2128 - val_accuracy: 0.9364
Epoch 12/20
28/28 [=====] - 53s 2s/step - loss: 0.0263 - accuracy: 0.9937 - val_loss: 0.2681 - val_accuracy: 0.9364
Epoch 13/20
28/28 [=====] - 62s 2s/step - loss: 0.0670 - accuracy: 0.9758 - val_loss: 0.1716 - val_accuracy: 0.9318
Epoch 14/20
28/28 [=====] - 58s 2s/step - loss: 0.0751 - accuracy: 0.9777 - val_loss: 0.1834 - val_accuracy: 0.9409
Epoch 15/20
28/28 [=====] - 69s 2s/step - loss: 0.0474 - accuracy: 0.9861 - val_loss: 0.1705 - val_accuracy: 0.9455
Epoch 16/20
28/28 [=====] - 56s 2s/step - loss: 0.0268 - accuracy: 0.9909 - val_loss: 0.1820 - val_accuracy: 0.9500
Epoch 17/20
28/28 [=====] - 53s 2s/step - loss: 0.0264 - accuracy: 0.9892 - val_loss: 0.1300 - val_accuracy: 0.9545
INFO:tensorflow:Assets written to: model-017.model\assets
Epoch 18/20
28/28 [=====] - 58s 2s/step - loss: 0.0258 - accuracy: 0.9939 - val_loss: 0.3044 - val_accuracy: 0.9182
Epoch 19/20
28/28 [=====] - 50s 2s/step - loss: 0.0198 - accuracy: 0.9972 - val_loss: 0.1633 - val_accuracy: 0.9545
Epoch 20/20

```

- Now as our model is trained, we will choose the model number with highest accuracy, in our case which is ‘model-017’. Now we load our model with model-017 file. After fitting the model, I used cascade classifier as ‘haarcascade\_frontalface\_default.xml’

```

In [32]: print(model.evaluate(test_data,test_target))
9/9 [=====] - 4s 413ms/step - loss: 0.2036 - accuracy: 0.9420
[0.20355169475078583, 0.9420289993286133]

In [33]: from keras.models import load_model

In [43]: model = load_model('model-017.model')

face_cascade=cv2.CascadeClassifier('C:/Users/KIIT/Desktop/haarcascade_frontalface_default.xml')

source=cv2.VideoCapture(0)

labels_dict={0:'NO MASK',1:'MASK'}
color_dict={0:(0,0,255),1:(0,255,0)}

In [44]: while(True):

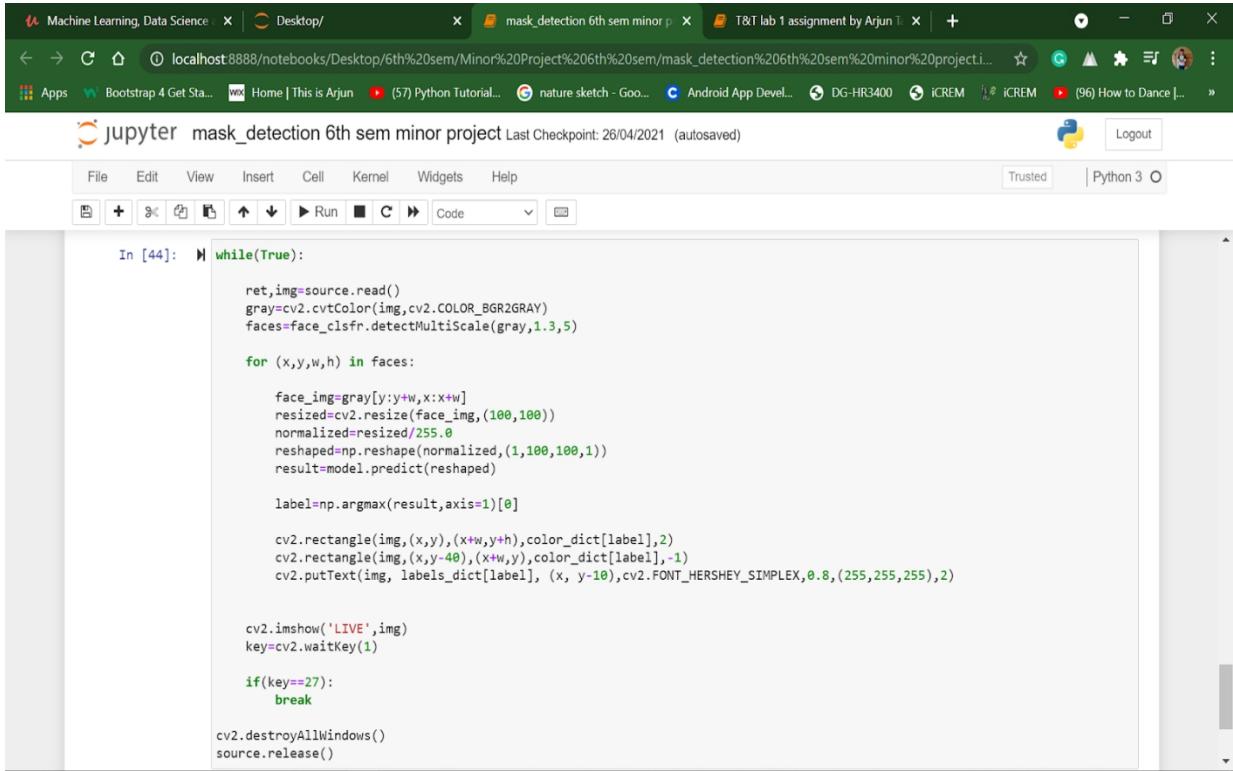
    ret,img=source.read()
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces=face_cascade.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:

        face_img=gray[y:y+w,x:x+w]

```

- Now it's time for our model to run. we give some rectangle configuration and colour for our rectangle; in our case they are red for 'no-mask' and green for 'mask'



The screenshot shows a Jupyter Notebook interface with the title "jupyter mask\_detection 6th sem minor project". The notebook has a single cell (In [44]) containing Python code for face mask detection using OpenCV and a pre-trained model. The code reads an image, converts it to grayscale, detects faces, normalizes them, reshapes the data, and uses a model to predict the mask status. It then draws rectangles around the faces and text labels indicating the result ('LIVE' or 'NO MASK'). The code also includes a key press loop to exit.

```
In [44]: while(True):
    ret,img=source.read()
    gray=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces=face_clsfr.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        face_img=gray[y:y+w,x:x+w]
        resized=cv2.resize(face_img,(100,100))
        normalized=resized/255.0
        reshaped=np.reshape(normalized,(1,100,100,1))
        result=model.predict(reshaped)

        label=np.argmax(result,axis=1)[0]

        cv2.rectangle(img,(x,y),(x+w,y+h),color_dict[label],2)
        cv2.rectangle(img,(x,y-40),(x+w,y),color_dict[label],-1)
        cv2.putText(img, labels_dict[label], (x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)

    cv2.imshow('LIVE',img)
    key=cv2.waitKey(1)

    if(key==27):
        break

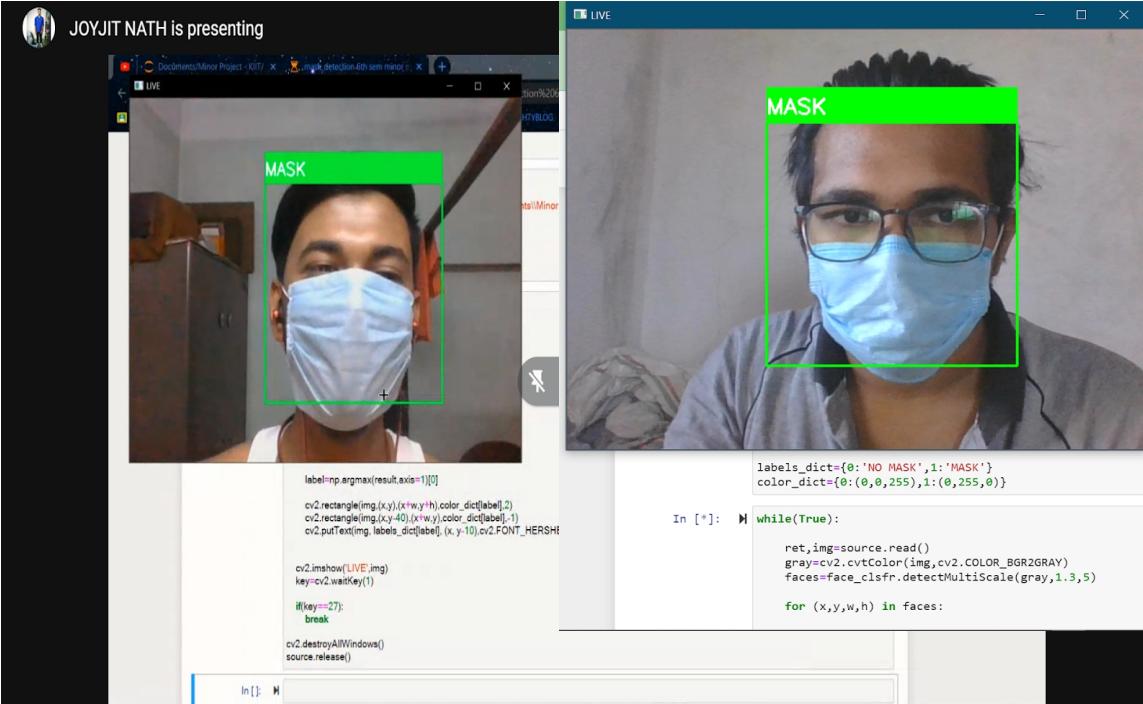
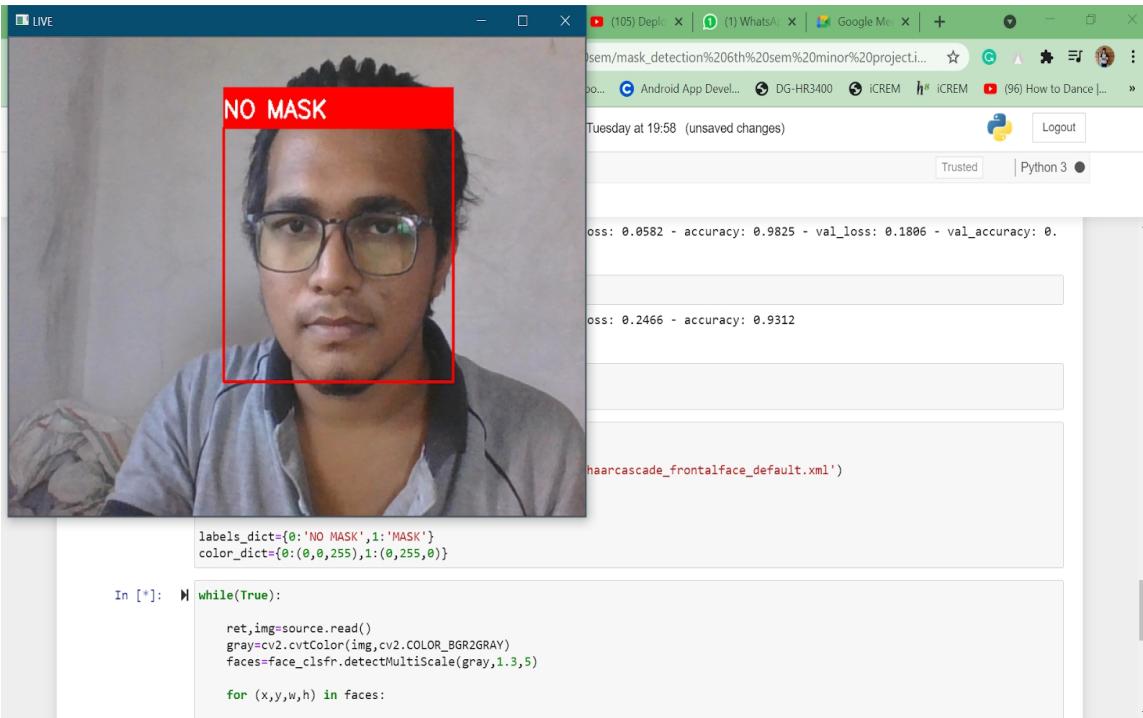
cv2.destroyAllWindows()
source.release()
```

And it's done.

## CHAPTER 4

### RESULTS & DISCUSSIONS

we have attached the screenshots of our results, hereby we can say that our model is successful.



## CHAPTER 5

### CONCLUSION AND FUTURE WORK

#### 5.1 Conclusion

To mitigate the spread of COVID-19 pandemic, measures must be taken. We have modeled a facemask detector using SSD architecture and transfer learning methods in neural networks. To train, validate and test the model, we used the dataset that consisted of 1916 masked faces images and 1919 unmasked faces images. These images were taken from various resources like Kaggle and RMFD datasets. The model was inferred on images and live video streams. To select a base model, we evaluated the metrics like accuracy, precision and recall and selected architecture with the best performance having 100% precision and 99% recall. It is also computationally efficient which makes it easier to use it. This face mask detector can be deployed in many areas like shopping malls, airports and other heavy traffic places to monitor the public and to avoid the spread of the disease by checking who is following basic rules and who is not.

#### 5.2 Future Work

More than fifty countries around the world have recently initiated wearing face masks compulsory. People have to cover their faces in public, supermarkets, public transports, offices, and stores. Retail companies often use software to count the number of people entering their stores. They may also like to measure impressions on digital displays and promotional screens. We are planning to improve our Face Mask Detection tool and release it as an open-source project. Our software can be equated to any existing USB, IP cameras, and CCTV cameras to detect people without a mask. This detection live video feed can be implemented in web and desktop applications so that the operator can see notice messages. Software operators can also get an image in case someone is not wearing a mask. Furthermore, an alarm system can also be implemented to sound a beep when someone without a mask enters the area. This software can also be connected to the entrance gates and only people wearing face masks can come in.

**REFERENCES –**

<https://www.python.org/>

[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

[youtube.com](http://youtube.com)

## SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

### AI MASK DETECTION MODEL

ARJUN TANPURE

1829203

**Abstract:** Our objective towards the model was to create a software which can help health care personnel to reduce covid spread chain. and according to me our model will help people realize that with the help of technology we can contribute in making a better future for the human race.

**Individual contribution and findings:** After the data cleaning and adding the libraries, after loading the model and testing the model, I came to the conclusion of using the model 017, as its accuracy was the best one. after fitting the model, I used cascade classifier as 'haarcascade\_frontalface\_default.xml' after that I have given rectangle specifications and colour to it, finally we run our model and it was successful. The entire project was completed by both of us.

**Individual contribution to project report preparation:** Abstract and chapters are written by me and all others are completed by my teammate.

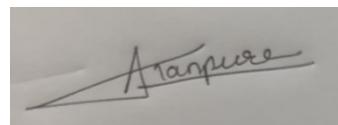
So finally, I would say our contribution to the project is equal in all aspects, weather it is training the model or presenting to teacher or writing report, we both have same contribution.

**Individual contribution for project presentation and demonstration:** As we are only two people in group, we both have presented equally.

Full Signature of Supervisor:

.....

Full signature of the student:



**JOYJIT NATH**

**1829156**

**Abstract:** Our objective towards the model was to create a software which can help health care personnel to reduce covid spread chain. and according to me our model will help people realize that with the help of technology we can contribute in making a better future for the human race.

**Individual contribution and findings:** At first, I have researched and studied about all the libraries used in python for neural networks Then I have imported the main libraries used for our model, the libraries I used was keras, NumPy, OS, etc. I found the best dataset suitable for our model from Kaggle. after importing the dataset, I categorized the data set, I had also helped my team mate train the model and also tested the model.

**Individual contribution to project report preparation:** The entire project and the report was executed and completed by both of us through google meetings. My individual contribution in report is I have structured the entire report and also the acknowledgement to the teacher. the chapters were written by both of us on google meet.

So finally, I would say our contribution to the project is equal in all aspects, weather it is training the model or presenting to teacher or writing report, we both have same contribution.

**Individual contribution for project presentation and demonstration:** As we are only two people in group, we both have presented equally.

Full Signature of Supervisor:

.....

Full signature of the student:

Joyjit Nath