**FACE MASK DETECTION USING MACHINE LEARNING**

**-*An Approach to Reduce Risk of Corona Virus Spread***

**A Mini Project Report Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of**

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**By**

**NADIMINTI AKASH 19PA1A04B1**

**NAKKA VENKATA DATTA VENU 19PA1A04B3**

**LALAM SAI SUBASH 19PA1A0489**

**KAISARLA SUJANA 19PA1A0468**

Under the Esteemed Guidance of

**Dr.Argha Sarkar**

**Associate Professor**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**VISHNU INSTITUTE OF TECHNOLOGY (A)**

**(Accredited by NBA, NAAC, Approved by AICTE & Affiliated to JNTU Kakinada) Vishnupur, Bhimavaram - 534202.**

**2021 - 2022**

**VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS)**

**(Accredited by NBA, NAAC, Approved by AICTE & Affiliated to JNTU Kakinada) Vishnupur, Bhimavaram-534202**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**



**CERTIFICATE**

This is to certify that the Mini Project entitled **“FACE MASK DETECTION USING MACHINE LEARNING –*AN APPROACH TO REDUCE CORONA VIRUS SPREAD*”** is being submitted by **NADIMINTIAKASH(19PA1A04B1)**,**KAISARLA SUJANA(19PA1A0468),LALAM SAI SUBASH(19PA1A0489),NAKKA VENKATA DATTA VENU(19PA1A04B3),** in partial fulfillment for the award of the degree of **Bachelor of Technology in Electronics and Communication Engineering** is a record of the bonafide work carried out by them under my guidance and supervision during academic year 2021–2022 and it has been found worthy of acceptance according to the requirements of the university.

**Dr.Argha Sarkar Dr.N.Padmavathy**, PhD (IITKGP)

**Associate Professor, ECE Professor & Head, ECE**

**Project Guide**

**External Examiner**

## ABSTRACT

In recent times, due to the vigorous transmission of COVID-19 SARS virus from one person to another person, it I made necessary by the World Health Organization to use face maks.Inorder to avoid the transmission of virus, all the people should wear masks and those who are violated can be penalized. To penalize, we should detect if the person is wearing the mask or if the person is not wearing the mask. This can be done using Open-cv python and some classification models like SVM classifier. This project is implemented in live. If a person wears a mask, it will show mask or it will show no mask. So using this face mask detection, we can reduce the spread of COVID-19 virus in the public places like banks, malls e.t.c.

### Keywords: COVID-19, Open-cv, SVM, SARS, Face mask

## CONTENTS

### Topic Page No

[ABSTRACT iii](#_bookmark0)

**1. INTRODUCTION 2-4**

**2. LITERATURE SURVEY 5-8**

**3. DATASET DESCRIPTION 9-11**

**3.1 SOURCE DATASET 10**

**3.2 SIZE OF DATASET 10**

**3.3 HIERARCHY OF DATASET 10**

**3.4 SAMPLE DATASET 11**

4. DATA PRE-PROCESSING 12-14

4.1 ARRAY TO IMAGE CONVERTION 13

4.2 PRE PROCESSING INPUT 13

4.3 DATA GENERATION 13

4.4 DATA SAVING FILES 14

5. PROPOSED MODEL 15-17

5.1 CASCADE CLASSIFIER 16

5.2 SUPPORT VECTOR MACHINE 16-17

6.EXPERIMENTAL SETUP 18-19

6.1 SPLITTING THE DATA 19

6.2 CREATING THE MODEL 19

7.EVALUATING THE MODEL 20-22

7.1 EVALUATION METRICS 21

7.2 EVALUATION RESULTS 21

7.3 TESTING 22

8.CONCLUSION 23-24

REFERENCES 25-26

APPENDIX 27-31

**CHAPTER 1**

**INTRODUCTION**

## 1. INTRODUCTION

**1.1 Motivation:**

In 2019,this COVID-19 was declared as pandemic by WHO.If the pandemic is not controlled it would wipe out the face of earth.Hence there is a strong need to put collective efforts in order to reduce the spread of spread of Covid-19 virus as soon as possible.A series of safety protocols were issued by the WHO.All the countries and their governments were ordered to follow all the protocols to prevent the spreading of virus further.Those protocols include wearing mask,sanitizing hands,maintain social distance e.t.c.A part of the contribution is to ensure that everyone wear maks using the technology around.the results from our project will be used to ensure that everyone is wearing mask or not.We can penalize the people who not following this protocol.



Fig 1.1: COVID-19 Pandemic

**1.2 Thought behind the work:**

Face Mask is necessary foer the people to stop the spread of virus.So,to make sure whether all the people are wearing mask,there is a need to detect the face mask.This can be done using machine learning techniques.In this machine learning lot of techniques like SVM,Regression,Classification,PCA,k-map model e.t.c.Within all these models we used SVM classifier to communicate rapidly and it has less parameters compared to others.Using these high end technologies,we can easily predict the face mask.By this prediction,we can avoid the spread.

* 1. **Problem Statement:**

In the present situations, people are not wearing masks in the public places like shopping malls, banks etc.To overcome this problem, we came up with an idea to use a detector that clicks the pictures of human beings continuously.It detects from their faces whether they are wearing a mask or not without manual involvement.This detector is helpful to detect the people faces using image processing and machine learning**.**

* 1. **Hardware Requirements:**

RAM : Minimum 8GB

Processor : Intel Core i5 or more

* 1. **Software Requirements:**

Programming Language : Python 3

Programming Interface : Anaconda with Jupyter notebook

# 

# 

# 

# CHAPTER 2

**LITERATURE SURVEY**

## 2. LITERATURE SURVEY

**2.1 Brief Literature survey:**

The COVID-19 pandemic is causing a worldwide emergency in healthcare. This virus mainly spreads through droplets which emerge from a person infected with coronavirus and poses a risk to others. The risk of transmission is highest in public places. One of the best ways to stay safe from getting infected is wearing a face mask in open territories as indicated by the World Health Organization (WHO). In this project, we propose a method which employs TensorFlow and OpenCV to detect face masks on people. A bounding box drawn over the face of the person describes weather the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing face mask and an email will be sent to that person warning them that they are not wearing a mask so that they can take precautions.

Deep learning is a subfield of machine learning and deep neural architectures can extract high level features automatically without handcraft feature engineering unlike traditional machine learning algorithms. In this paper, we propose a method, which combines feature extraction layers of a convolutional neural network with traditional machine learning algorithms, such as, support vector machine, gradient boosting machines, and random forest. All of the proposed hybrid models and the above mentioned machine learning algorithms are trained on three different datasets: MNIST, Fashion-MNIST, and CIFAR-10. Results show that the proposed hybrid models are more successful than traditional models while they are being trained from raw pixel values. In this study, we empower traditional machine learning algorithms for classification using feature extraction ability of deep neural network architectures and we are inspired by transfer learning methodology to this.

COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion. The method attains accuracy up to 95.77% and 94.58% respectively on two different datasets. We explore optimized values of parameters using the Sequential Convolutional Neural Network model to detect the presence of masks correctly without causing over-fitting.

**2.2 OpenCV:**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

**2.3 Numpy:**

NumPy is an open source project aiming to enable numerical computing with Python. It was created in 2005, building on the early work of the Numeric and Numarray libraries. NumPy will always be 100% open source software, free for all to use and released under the liberal terms of the modified BSD license.NumPy is developed in the open on GitHub, through the consensus of the NumPy and wider scientific Python community.

**2.4 Matplotlib:**

Matplotlib is a comprehensive library for creating static,animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

* Create publication quality plots.
* Make interactive figures that can zoom, pan, update.
* Customize visual style and layout.
* Export to many file formats.
* Embed in JupyterLab and Graphical User Interfaces.
* Use a rich array of third-party packages built on Matplotlib.

**2.5 Principal Component Analysis:**

Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance. Finding such new variables, the principal components, reduces to solving an eigenvalue/eigenvector problem, and the new variables are defined by the dataset at hand, not *a priori*, hence making PCA an adaptive data analysis technique. It is adaptive in another sense too, since variants of the technique have been developed that are tailored to various different data types and structures. This article will begin by introducing the basic ideas of PCA, discussing what it can and cannot do. It will then describe some variants of PCA and their application

# 

# 

# CHAPTER 3

# DATASET DESCRIPTION

## DATASET DESCRIPTION

**3.1 Source of dataset:**

In this model,we have created the dataset using open cv and haar\_data classifier.With this classifier we created the dataset using realtime face images from the system camera and saved the images in a numpy file.With the help of these datasets we created the model and the real time model.

**3.2 Hierarchy of dataset:**

Dataset is classified into two sub-types i.e with mask and without mask.

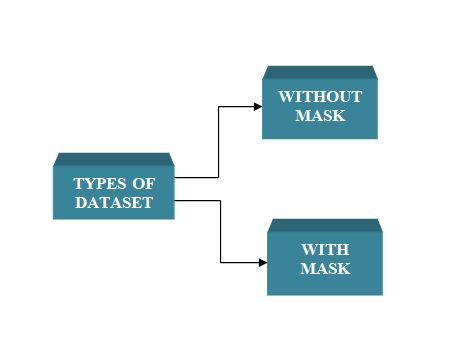


Fig 3.1 Hierarchy of the dataset

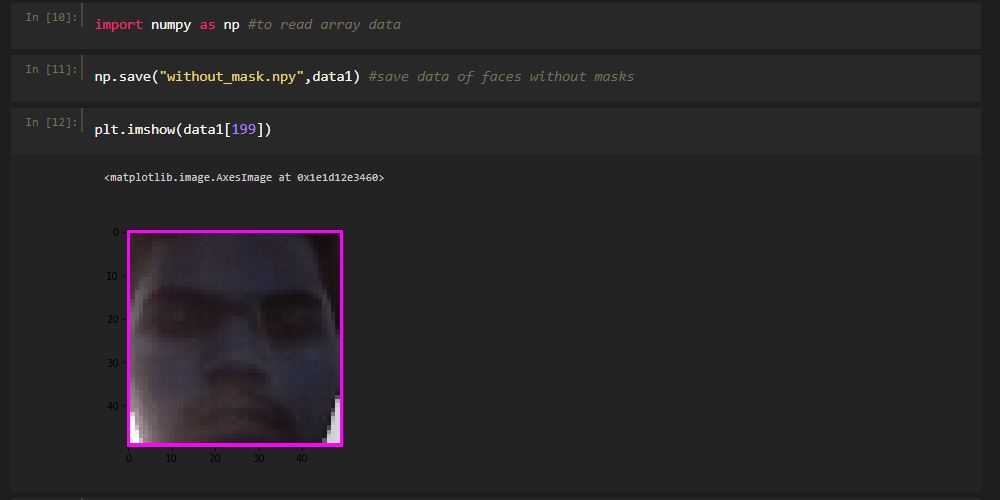
**3.3 Size of dataset:**

The collected dataset contain 2000 images which of size nearly 250 Mb.

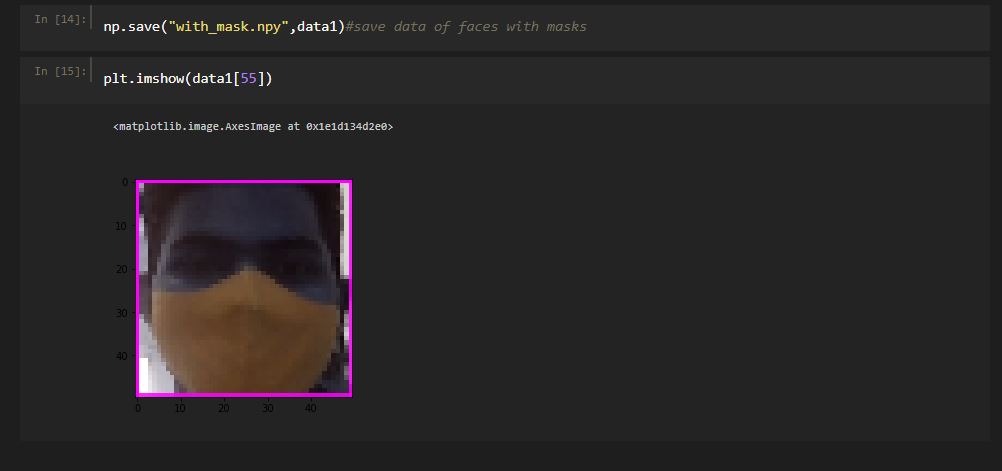
|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Name of the folder | Size | Number of images |
| 1 | With Mask | 150mb | 1000 |
| 2 | Without Mask | 100mb | 1000 |

Fig 3.2 Size of dataset

**3.4 Sample Dataset:**



Without Mask Sample



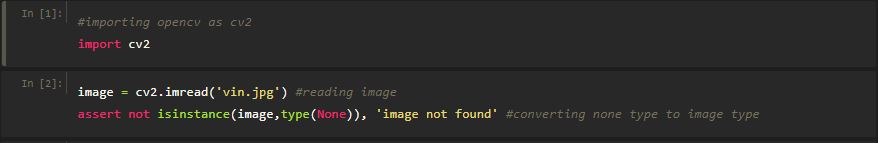
With Mask Sample

# CHAPTER 4

**DATA PRE PROCESSING**

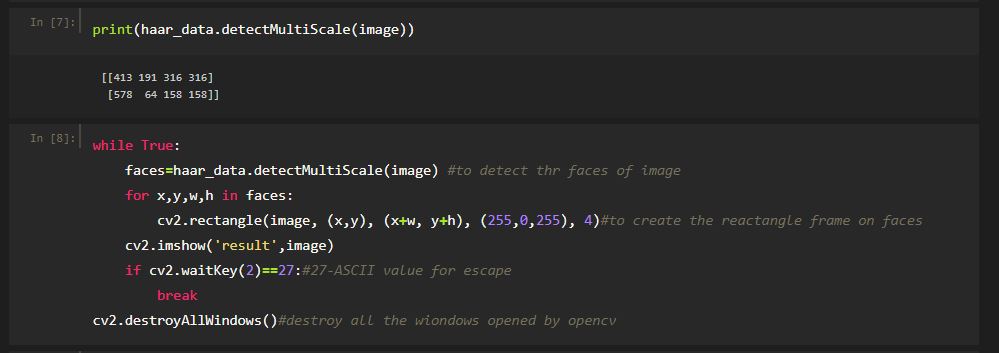
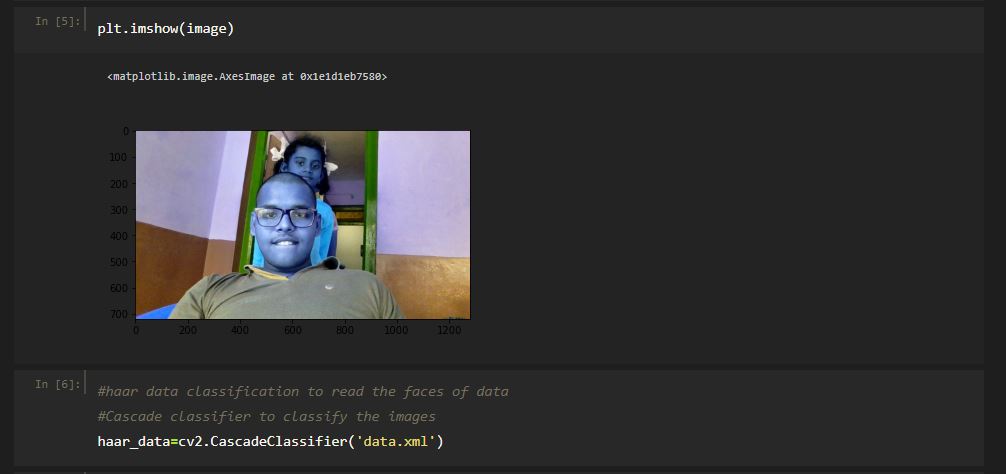
## 4. DATA PRE-PROCESSING

**4.1 ARRAY TO IMAGE CONVERTION:**

****

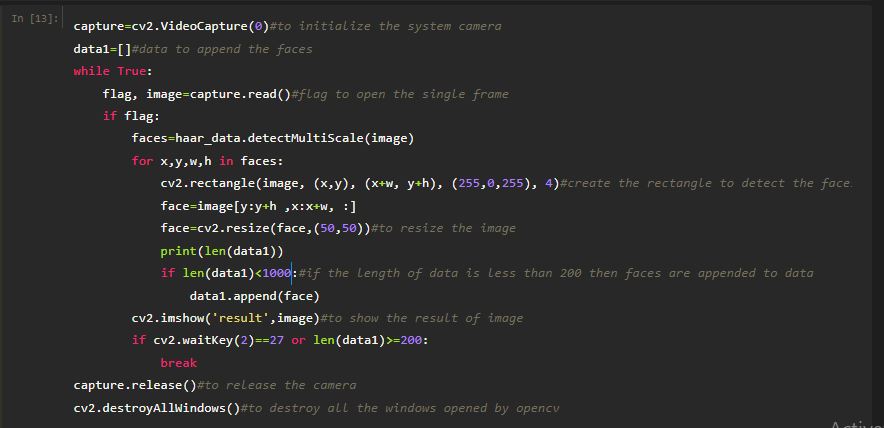
# The above code converts array to image.

**4.2 PRE-PROCESSING INPUT:**

****

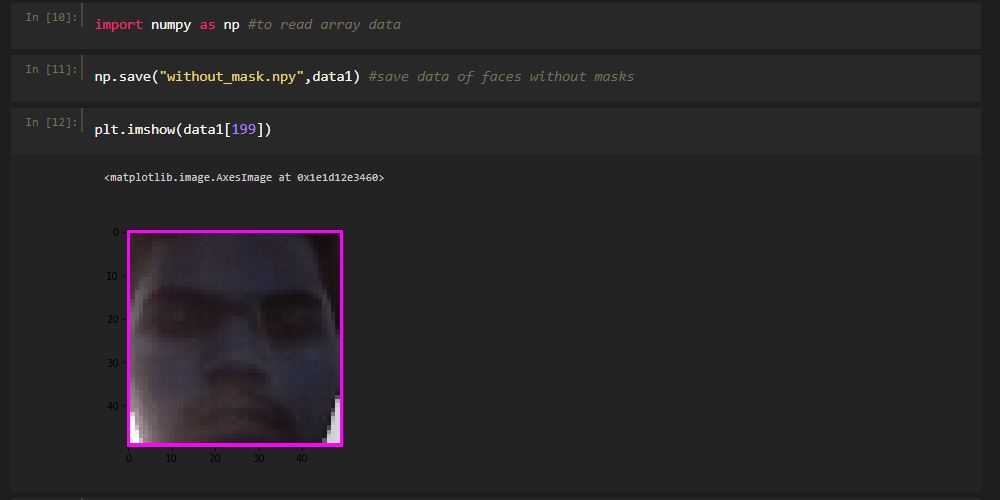
The above blocks are written to detect the faces images using haar\_data classifier and to draw a frame on the face.

**4.3 Data Generation:**

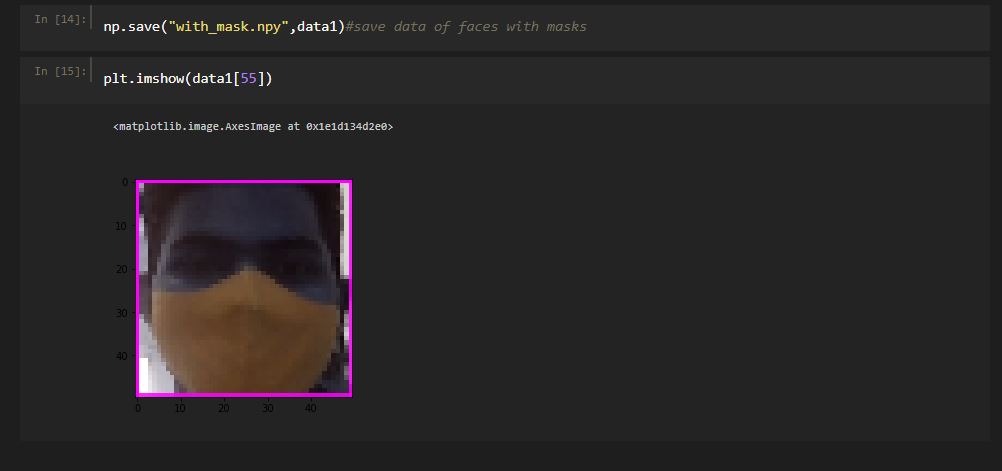
****

The above block generates the data of images and append them in the data 1 list.It will close the camera when the size of data 1 reach to 1000 images.

**4.4 Data saving Files:**

****

Without mask dataset

****

With mask dataset

**CHAPTER 5**

**PROPOSED MODEL**

**5.PROPOSED MODEL**

**5.1 CASCADE CLASSIFIER:**

Haar feature-based cascade classifiers is an effectual machine learning based approach, in which a cascade function is trained using a sample that contains a lot of positive and negative images. The outcome of AdaBoost classifier is that the strong classifiers are divided into stages to form cascade classifiers. The term “cascade” means that the classifier thus produced consists of a set of simpler classifiers which are applied to the region of interest until the selected object is discarded or passed.

The cascade classifier splits the classification work into two stages: training and detection. The training stage does the work of gathering the samples which can be classified as positive and negative. The cascade classifier employs some supporting functions to generate a training dataset and to evaluate the prominence of classifiers.

In order to train the cascade classifier, we need a set of positive and negative samples. In our work, we have incorporated the utility called opencv createsample to create the positive samples for opencv\_traincascade. The output file of this function serves as an input to opencv traincascade to train the detected face. The negative samples are collected from arbitrary images, which do not include the objects to be detected.

**5.2 SUPPORT VECTOR MACHINE:**

**Support vector machines (SVMs)** are a set of supervised learning methods used for [classification](https://scikit-learn.org/stable/modules/svm.html#svm-classification), [regression](https://scikit-learn.org/stable/modules/svm.html#svm-regression) and [outliers detection](https://scikit-learn.org/stable/modules/svm.html#svm-outlier-detection).

The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

* If the number of features is much greater than the number of samples, avoid over-fitting in choosing [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) and regularization term is crucial.
* SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

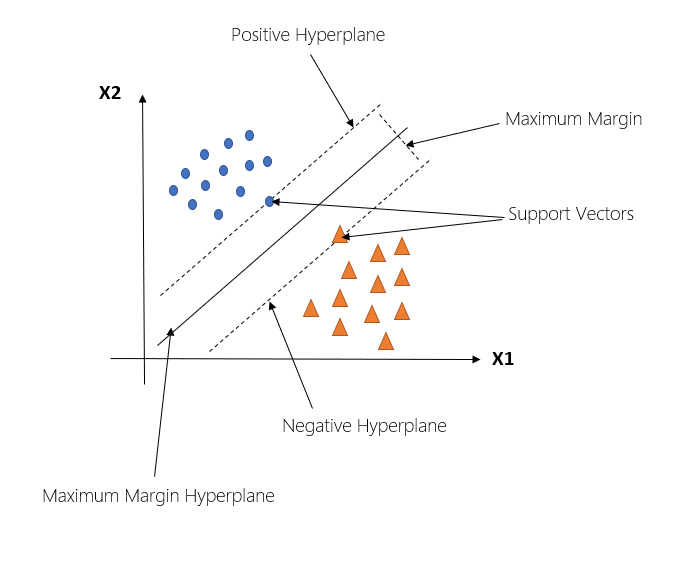
****

Fig 5.1 SVM Hyperplane separating two different classes

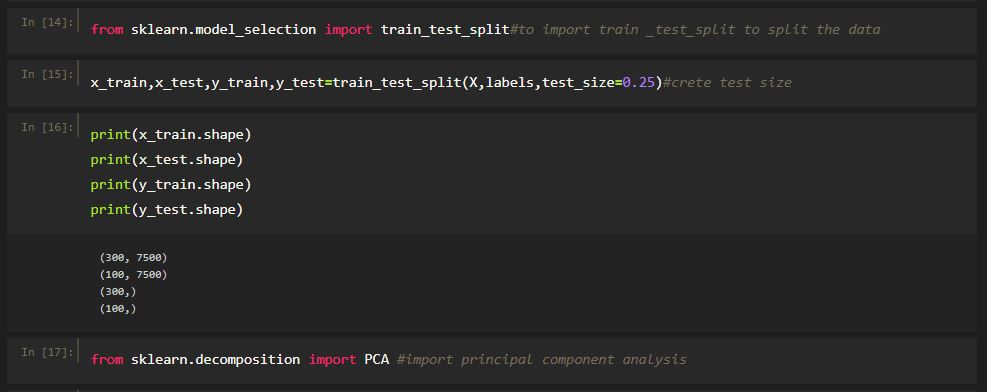
* The above figure shows the hyperplane formed by the support vectors using SVM.

**CHAPTER 6**

**EXPERIMENTAL SETUP**

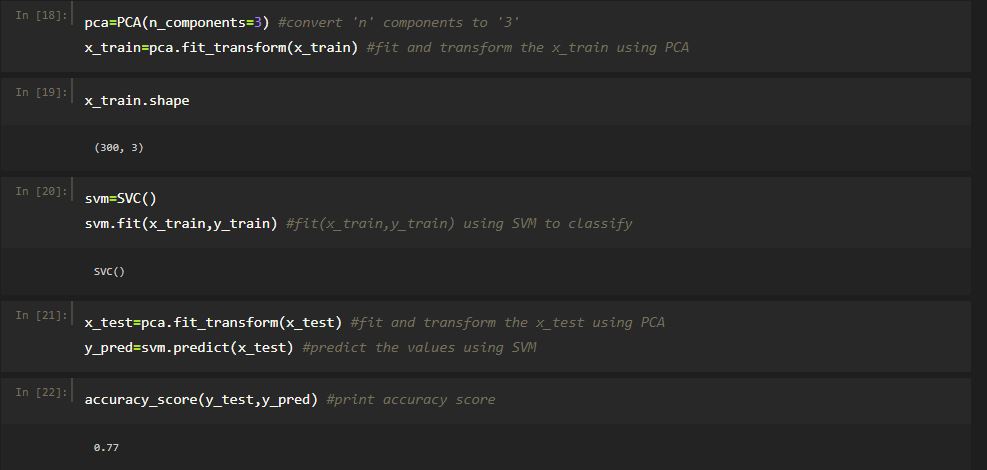
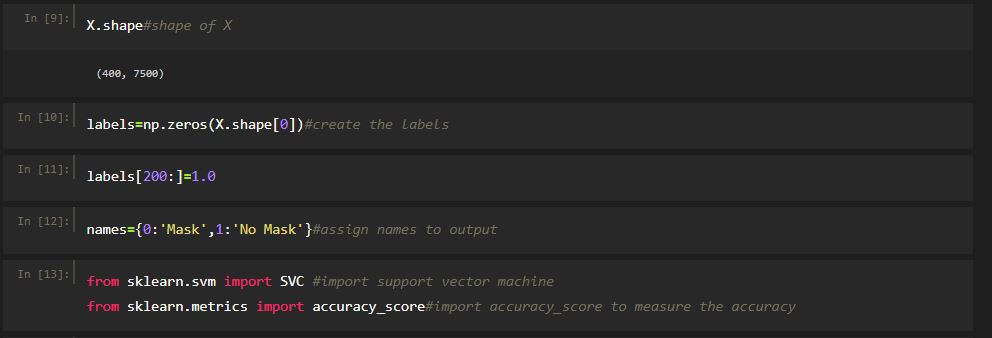
**6.EXPERIMENTAL SETUP**

**6.1 Splitting the data:**

****

We splitted the data into training data and testing data to train and test the model and find the accuracy of the model.

**6.2 Creating the model:**

****

The model is created used SVM and fitted the data using the principal component analysis.

**CHAPTER 7**

**EVALUATING THE MODEL**

**7.EVALUATING THE MODEL**

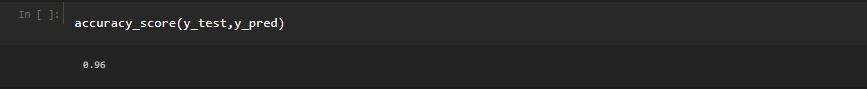
Model evaluation is an integral part of the model development process.It helps to find the best model that represents our data and how well the model will work in future.Evaluating the model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic models.

**7.1 Evaluating Metrics:**

The evaluation of a model is performed with the help of a confusion matrix.Totally,four outcomes are generated by a confused matrix, namely TP,TN,FP,FN.The measures used for the calculation are as follows:

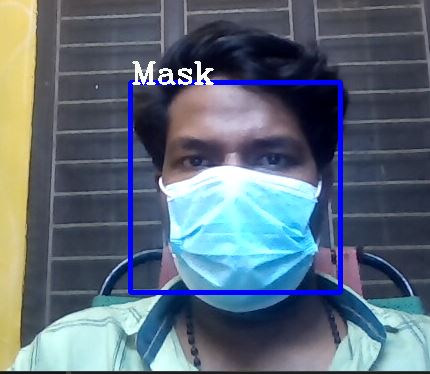
* Accuracy
* Sensitivity
* Specificity
* Precision
* F-Score

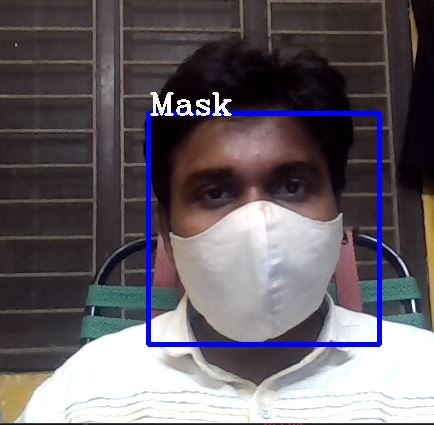
**7.2 EVALUATING RESULTS:**

****

Model Accuracy is 0.96

**7.3 Testing:**

****

****

* The above pictures show us the results from the system whether the people are wearing mask or not.

**CHAPTER-8**

# CONCLUSION

**8.CONCLUSION**

Our effort behind this project is to design and fabricate a gadget which is so compact in itself that provides an advantage of reducing corona virus spreading around the world.For this we built a model of face mask detection system which can be used at public places like shopping malls, banks and other crowded places etc.This realtime face mask detector can be implanted anywhere and we can penalize people by recognizing their faces.This model can be installed within the devices with minimum RAM and it is low cost reliable.This model can also be implemented using some microcontroller devices and IOT devices,using the above devices it can be made more preferable to people to use in their hands. With this we are looking forward to reduce the spread of corona virus

**REFERENCES**

[1] Das, Arjya, Mohammad Wasif Ansari, and Rohini Basak. "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV." 2020 IEEE 17th India Council International Conference (INDICON). IEEE, 2020.

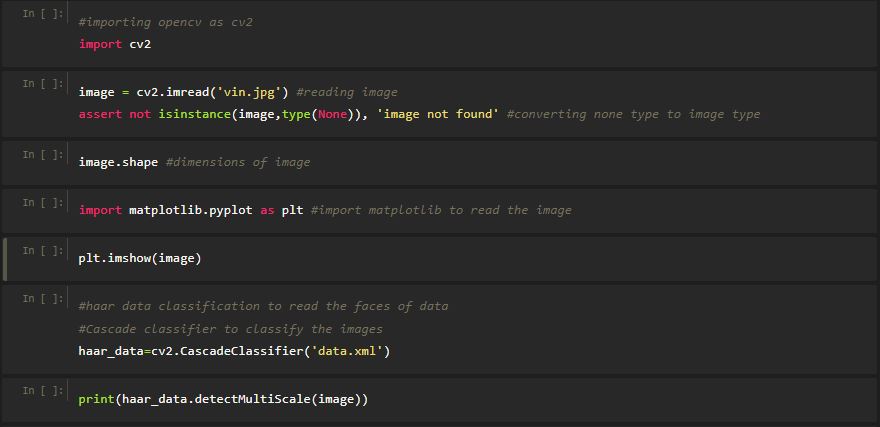
[2] "Feature extraction based on deep learning for some traditional machine learning methods." 2018 3rd International Conference on Computer Science and Engineering (UBMK). IEEE, 2018.

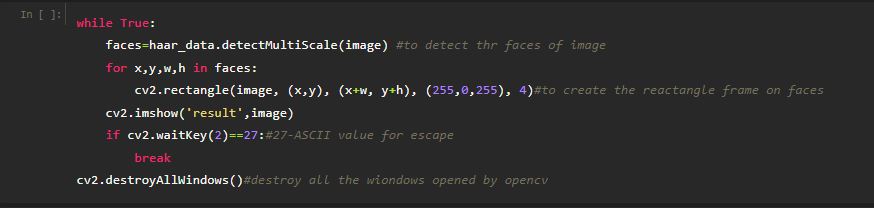
[3] Loey, Mohamed, et al. "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic." Measurement 167 (2021): 108288.

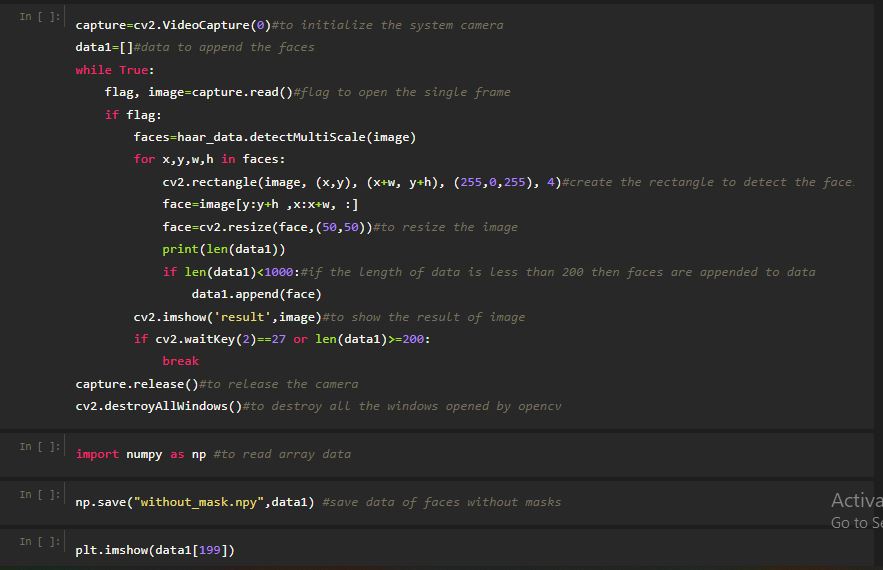
[4] Adusumalli, Harish, et al. "Face Mask Detection Using OpenCV." 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). IEEE, 2021.

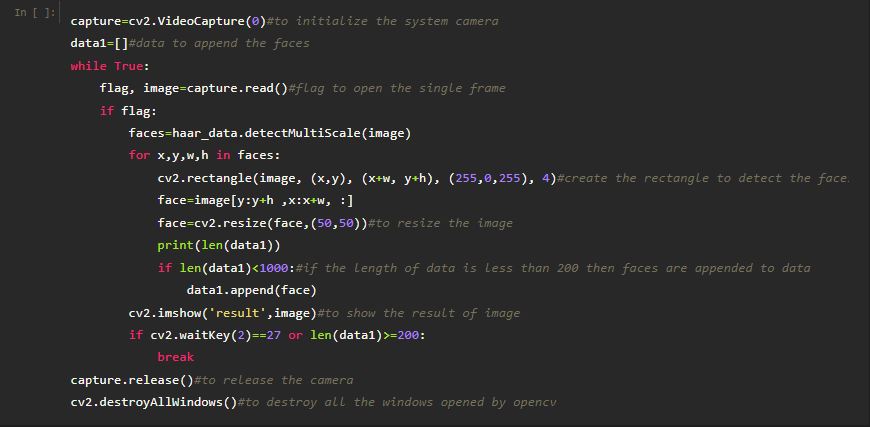
**APPENDIX**

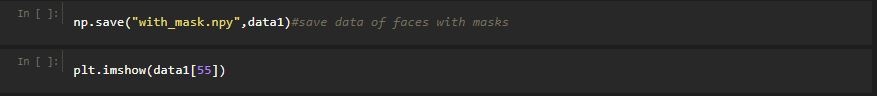
**CODE**

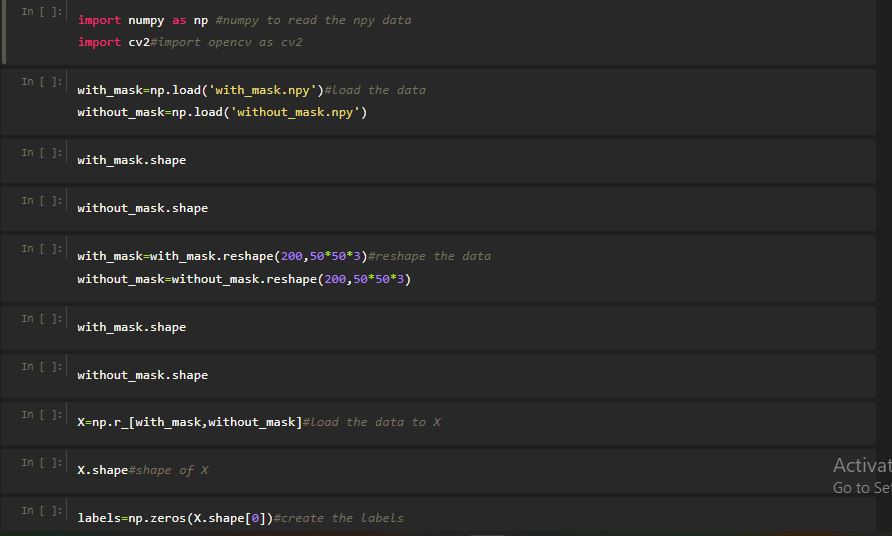
****

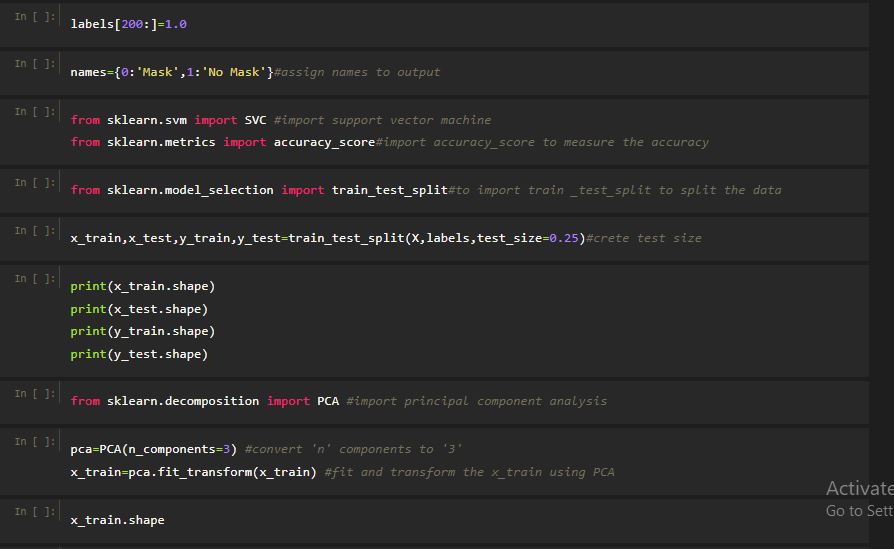
****

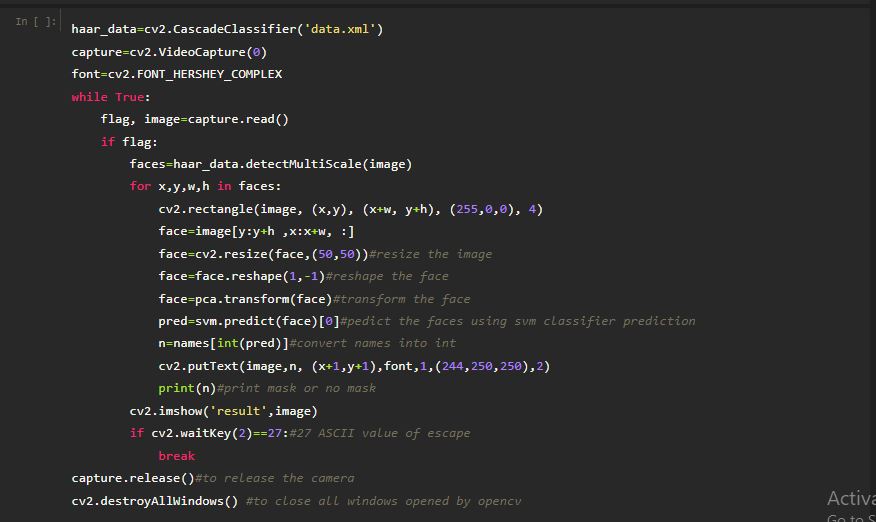
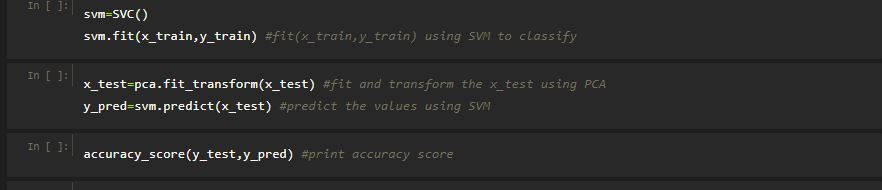
****

****

****

****

****

****