MIT Academy of Engineering

School of Computer Engineering And Technology
Presentation for Computer Vision Project

Real-Time Face Mask Detection System

By

Seat No.	Name	Roll No.
B224129	Ritesh Kulkarni	260
B224135	Rushikesh Jadhav	270
B224147	Omkar Shinde	272
B224152	Srushti Jadhav	284

Guide
Mrs. S. B. Kharat

Contents

1. Problem Statement
2. Motivation
3. Abstract
4. Motivation
5. Objectives
6. Methodology
7. Libraries Used
8. Result Screenshots
9. Conclusion
10. Necessary Links

Problem Statement

Design & Develop Real Time Face Mask Detection System .

Motivation

Today it has become mandatory for all the citizens to wear a face mask to protect themselves from COVID-19. This application can be helpful for all the shop owners, offices, banks or any public place because if anyone is not wearing a mask then he or she must not be allowed in that area. So, to take care of this problem we don’t need any guard or person who keeps a watch on people. We can integrate a camera which continuously clicks pictures of humans and detect from there faces whether they are wearing a face mask or not.

Abstract

- After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one.
- The basic aim of the project is to detect the presence of a face mask on human faces on live streaming video as well as on images.
- Real Time Face Mask Detection system is expected for checking that people faces are masked in regulated areas or not.
- Face detection is a key area in the field of Computer Vision and Pattern Recognition.
- To perform this task, a large dataset of masked faces is necessary for training deep learning models towards detecting people wearing masks and those not wearing masks.
- Besides, many people are not correctly wearing their masks due to bad practices, bad behaviors or vulnerability of individuals (e.g., children, old people).

Objectives

- Capturing Live Photographs(With 7 Without Mask) For Training The Machine Learning Model On Real Time Datasets.
- Preprocessing Of Data & Creating Data Directory For Training Model Using Opencv & Basic Python Libraries Like Numpy.
- Using Harrcascade For Face Detection In Image & Video.
- Training The Model With Proper Train-Test Split With the Use of PCA & SVA Machine Learning Algorithms.
- Capturing Live Video of User And Detecting Whether The User is Wearing Mask or Not Using Model & Opencv Features And Giving Alert According To It.

Methodology

- To install OpenCV open command prompt or terminal and type -
- Installation – pip install opencv-python
- Importing OpenCV first read an image
- Read any image using OpenCV it returns object of numpy array by default and using img.shape we are checking the height and width of image and also it returns 3 which is the color channel of image.
- The values of array are the color values actually
- Face detection algorithm was introduced by Viola and Jones in 2001. They divided this algorithm in four stages : 1. Haar Features Selection 2. Integral Images 3. AdaBoost 4. Cascading Classifier
- A XML file which is going to help us to detect faces from the image
- Returns x, y, width and height of the face detected in the image. We can draw a rectangle on the face.
- Iterate over the array returned to us by detectMultiScale method and put x,y,w,h in cv2.rectangle

Methodology

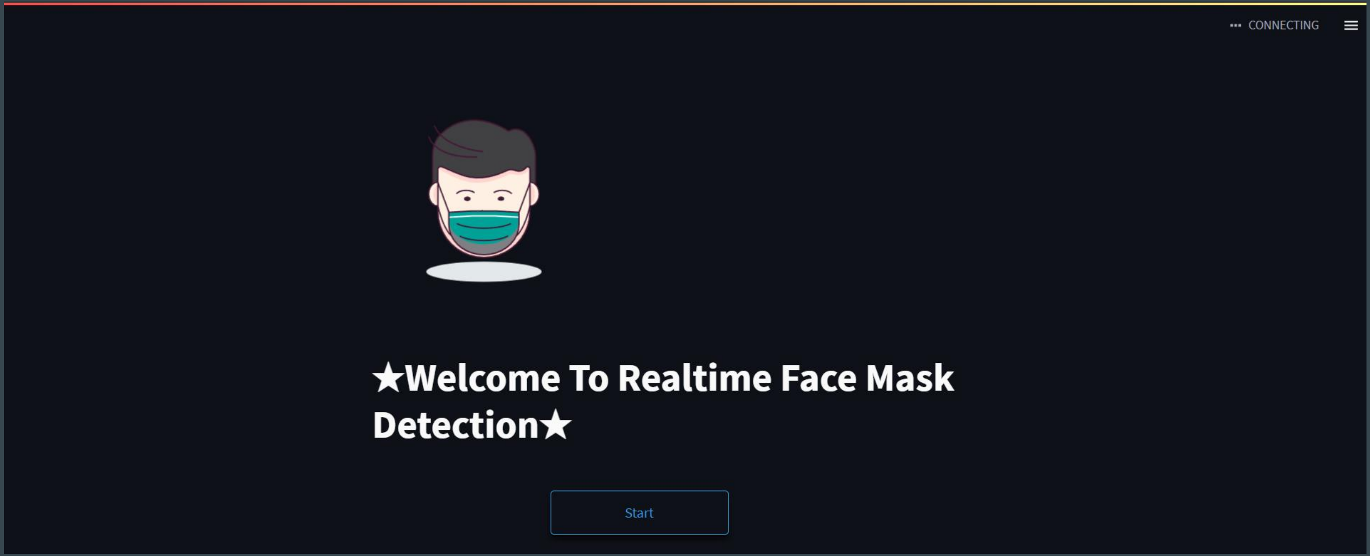
- We need images of people wearing a mask and not wearing a mask. We need to collect data and we are going to collect data using our own camera.
- Save the data in a numpy file and it can also plot the face data to check the data collected by OpenCV.
- Load the data anywhere and start processing it to apply machine learning on it.
- Data is loaded with the shape 200, 50, 50, 3. 200 is the number of images we have collected. 50, 50 is the size of each image. 3 is the color channel (red, green, blue). Reshape the data to make it 2D.
- Concatenate the data into a single array. NPR will help you to store data row wise.
- Create one array of zeros and assign first 200 indexes as zero and next 200 indexes as one. Because first 200 images belong to faces with mask and next 200 images belong to faces without mask.
- The algorithm we are using is SVM & PCA for Training Model.
- Finally, Detecting Face Mask & No Mask State In Video & Giving Alerst According to It.

Libraries Used

- Haarcascade (Face Detection)
- Sklearn(SVM (Support Vector Machine), PCA (Principal Component Analysis))
- Pygame(Beep Sound Alert)
- NumPy(Processing Image Data)
- Cv2(Capturing & Processing Image Data)
- Matplotlib(Display Image)

Resultant Outputs

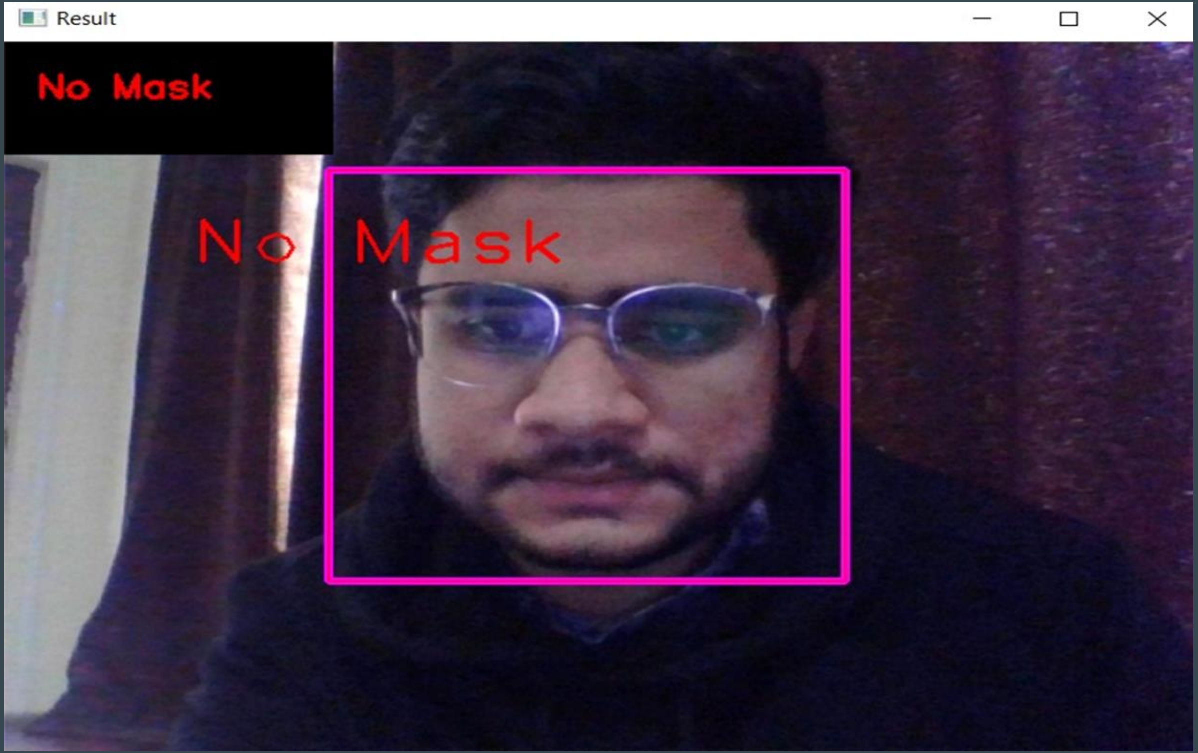
Streamlit (Localhost) Main Page



★Welcome To Realtime Face Mask Detection★

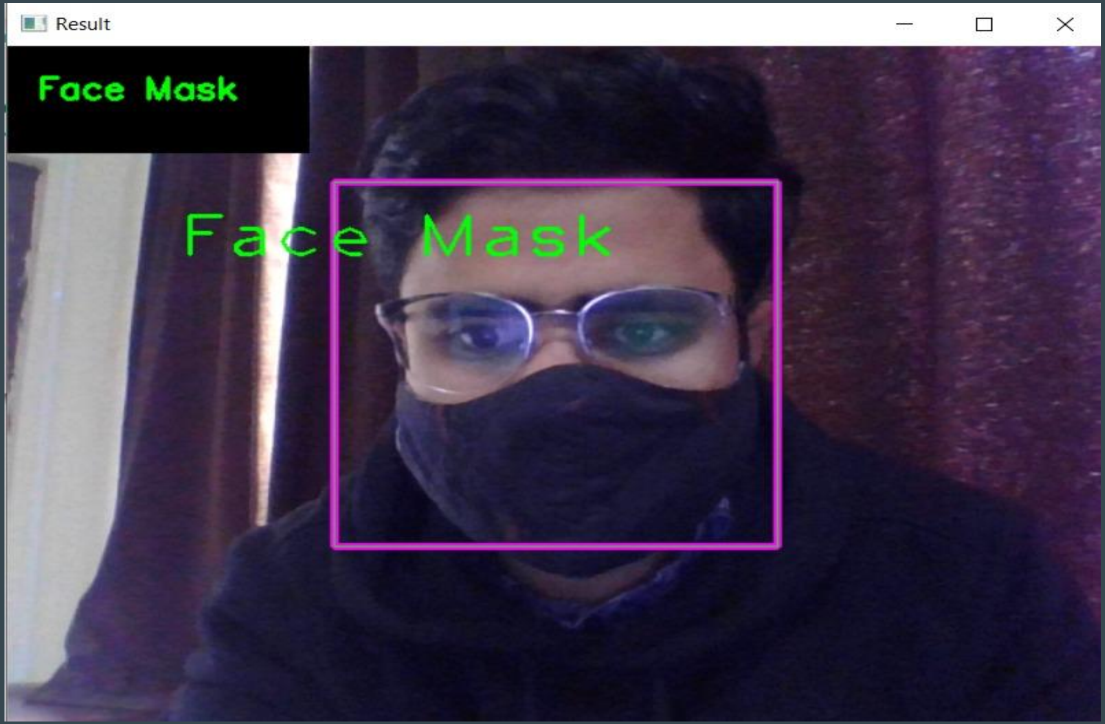
Resultant Outputs

No Mask
Detection
By System



Resultant Outputs

Face Mask Detection By System



Conclusion

- With the increasing number of COVID cases all over the world, a system to replace humans to check masks on the faces of people is greatly needed.
- This system satisfies that need.
- This system can be employed in public places like railway stations and malls.
- It will be of a great help in companies and huge establishments where there will be a lot of workers.
- This system will be of a great help there because it is easy to obtain and store the data of the employees working in that Company and will very easy find the people who are not wearing the mask.

Necessary Links

- GitHub Link :
https://github.com/RushiJadhav1507/Real-Time-Face-Mask-Detetction-System.git
- Google Drive Link :
https://drive.google.com/drive/folders/1pTUpIWrKeRwd6YisvMAvq-nFIIXxTlCf?usp=sharing

(As Our Project Uses pygame for beep alert & webcam for video, streamlit does not support both of services on deployment but our project runs correctly on streamlit localhost(use command >> steamlit run Project1.2.py))

Thank You!!!!