

**MINI PROJECT REPORT**

On

**FACE MASK DETECTION USING PYTHON**

THIRD YEAR OF ENGINEERING

By

**SAIF BODU**

**GANESH GHORPADE**

**SAAD SHAIKH**

Under the Guidance of

**PROF. MAHESH THAKUR**

Department of Computer Engineering



NAME OF DISSERTATION: **FACE MASK DETECTION USING PYTHON**

GROUP MEMBERS: **SAIF BODU**

**GANESH GHORPADE**

**SAAD SHAIKH**

CLASS: **THIRD YEAR COMPUTER ENGINEERING**

COLLEGE: **CHHATRAPATI SHIVAJI MAHARAJ INSTITUTE OF TECHNOLOGY**

SEMESTER: **VI**

NAME OF GUIDE: **PROF. MAHESH THAKUR**



CHHATRAPATI SHIVAJI MAHARAJ INSTITUTE OF TECHNOLOGY-410206 UNIVERSITY OF MUMBAI

Certificate

This is to certify that the project report entitled **“FACE MASK DETECTION USING PYTHON”** has been submitted by SAIF BODU, GANESH GHORPADE, and SAAD SHAIKH under the guidance of PROF. MAHESH THAKUR in partial fulfillment of the requirement for the award of third year of engineering in COMPUTER ENGINEERING from UNIVERSITY OF MUMBAI.

Certified by,

**PROF. MAHESH THAKUR**

PROJECT GUIDE

**INDEX**

|  |  |
| --- | --- |
| **Sr. no.** | **Name of topics** |
|  | Abstract |
|  | Introduction |
|  | Problem statement |
|  | Literature Survey |
|  | Existing System |
|  | Methodology |
|  | Proposed System |
|  | Hardware Requirement |
|  | Software Requirement |
|  | Source Code |
|  | Output |
|  | Conclusion |
|  | Reference |

**ABSTRACT**

The COVID-19 pandemic has caused major crisis with respect to the health of most of the human beings in most of the places of the world which leaves a greater impact by affecting the health of the people. Major effective methods are made by maintain social distancing and compulsorily wearing a mask. By wearing a mask, it mainly reduces the risk of transmission of the disease. We are trying to present a hybrid model using classical machine learning algorithms, deep learning for the detection. The dataset includes images with and without mask where we try to make use of OpenCV for the realtime detection using a webcam. As it is compulsory to wear face masks in public areas and for enhanced safety of people, we make sure to implement such systems for safety and security reasons. The same model can also be use in workplaces to ensure all the employees wear their masks throughout the day. We are using dataset to build a COVID-19 face mask detector using computer vision, TensorFlow, python and Keras. Our main goal is to identify whether a person on image/video stream is wearing a face mask or not with the help of deep learning and deep learning. As face recognition also represents the modalities of biometric.

**Keywords:** COVID-19, Deep learning, OpenCV, TensorFlow, Python, Keras.

**INTRODUCTION**

Because of the global COVID-19 corona virus outbreak, the wearing of face masks in public is becoming more common. People used to wear masks to protect their health from air pollution before Covid-19. Others, who are self-conscious about their appearance, mask their feelings from the public eye by covering their ears. COVID-19 transmission is clogged by wearing face masks, according to scientists. COVID-19 (also known as the corona virus) is the most recent epidemic virus to strike human health in the last century. COVID-19 has been declared a global pandemic by the World Health Organization (WHO) in 2020 due to its rapid spread.

COVID-19 infected over 5 million people in 188 countries in less than 6 months. The virus spreads by close contact and huddled, overcrowded environments. The corona virus outbreak has resulted in unprecedented levels of international scientific collaboration. Machine learning and Deep Learning, which are assisted by artificial intelligence (AI), can aid in the battle against Covid-19 in a variety of ways. Researchers and clinicians may use machine learning to predict the distribution of COVID-19 and use it as an early warning system for possible outbreaks. pandemics, as well as to identify and classify vulnerable groups. Aid distribution requires funding for emerging technologies such as artificial intelligence, IoT, big data, and machine learning in order to combat and forecast new diseases, better understand infection patterns, and trace and easily detect infections. AI's capacity is being used to combat the Covid-19 pandemic. In some countries, people are required by law to wear face masks in public. These rules and laws were created in response to the exponential increase in cases and deaths in a variety of areas. The method of tracking large groups of people, on the other hand, is becoming more complicated. Anyone who isn't wearing a face mask is detected during the observation process. We're going to implement a mask face recognition model that uses deep learning and computer vision. The proposed model could be used in conjunction with police work cameras to prevent COVID-19 transmission by detecting individuals who are not wearing face masks. The model was created using opencv, tensor flow, and keras, as well as deep learning and traditional machine learning techniques. For feature extraction, we combined deep transfer learning with three traditional machine learning algorithms. We made a comparison between them to find out which one is the best. within the method of coaching and detection, the required algorithmic rule that achieved the highest accuracy and consumed the smallest amount of time.

**PROBLEM STATEMENT**

The main objective of the face detection model is to detect the face of individuals and conclude whether they are wearing masks or not at that particular moment when they are captured in the image.

**LITERATURE SURVEY**

COVID-19 pandemic caused by novel coronavirus is continuously spreading until now all over the world. The impact of COVID-19 has been fallen on almost all sectors of development. The healthcare system is going through a crisis. Many precautionary measures have been taken to reduce the spread of this disease where wearing a mask is one of them. In this paper, we propose a system that restrict the growth of COVID-19 by finding out people who are not wearing any facial mask in a smart city network where all the public places are monitored with Closed-Circuit Television (CCTV) cameras. While a person without a mask is detected, the corresponding authority is informed through the city network. A deep learning architecture is trained on a dataset that consists of images of people with and without masks collected from various sources. The trained architecture achieved 98.7% accuracy on distinguishing people with and without a facial mask for previously unseen test data. It is hoped that our study would be a useful tool to reduce the spread of this communicable disease for many countries in the world.

**EXISTING SYSTEM**

face detection problem has been approached using Multi-Task Cascaded Convolutional Neural Network (MTCNN). Then facial features extraction is performed using the Google Face Net embedding model. 1.This system is capable to train the dataset of both persons wearing masks and without wearing masks. After training the model the system can predicting whether the person is wearing the mask or not wearing mask.

**METHODOLOGY**

Create a python environment test and active it

Load the Face Mask data

Pre-process the image using OpenCV

Generate & Train Face Mask classifier

Load Face Mask classifier from terminal

Load faces from Video stream

Apply the face mask detector to determine “Mask” Or “No Mask”

Result/Output

**Figure: Methodology of the project**

**PROPOSED SYSTEM**

1. This system is capable to train the dataset of both persons wearing masks and without wearing masks.
2. After training the model the system can predicting whether the person is wearing the mask or not.
3. It also can access the webcam and predict the result.

## HARDWARE REQUIREMENT

* Processor: Pentium IV
* 40 GB hard disk space.
* 256 MB RAM or more.
* 1.44 Floppy Disk Drive.
* 104 keys keyboard.
* Display capable of showing 65,000 colors or more.
* CD-ROM Drive for installing the package.
* Mouse with minimum two buttons.
* Web Cam.

## SOFTWARE REQUIREMENT

* Software: Spyder, VS-Code.
* OS : Windows / Linux / Solaris.

**SOURCE CODE**

**mainWindow.py**

import sys

import os

from tkinter import \*

from PIL import ImageTk, Image

#About us page

def about():

new\_window = Toplevel(root)

new\_window.geometry('600x600')

new\_window.title("About Us")

new\_window.resizable(False, False)

about\_label= Label(new\_window,text="About Us", height=1, bg="#03A9F4", fg="#ffffff")

about\_label.pack()

#creating tuple

Font\_tuples = ("Comic Sans MS", 20 ,"bold")

about\_label.configure(font= Font\_tuples)

frame= Frame(new\_window, width=500, height=400)

frame.pack()

frame.place(anchor='center',relx=0.5, rely=0.5)

label\_mem = Label(frame, text="GROUP MEMBERS:\n1. SAIF BODU\n2. GANESH GHORPADE\n3. SAAD SHAIKH\n\nCLASS:\nTHIRD YEAR COMPUTER ENGINEERING\n\nCOLLEGE:\nCHHTRAPATI SHIVAJI MAHARAJ INSTITUTE OF TECHNOLOGY\n\nNAME OF GUIDE:\nPROF. MAHESH THAKUR", font="Times 15" )

label\_mem.pack(padx=10,pady=10)

label\_ver = Label(frame, text="Version 1.0", font="Times 12" )

label\_ver.pack(padx=10,pady=10)

label\_copy = Label(frame, text="Copyrights ©2022 by SaifBodu", font="Times 12" )

label\_copy.pack(padx=10,pady=10)

frame.pack()

frame.place(anchor='center',relx=0.5, rely=0.5)

# Face mask detection page

def run():

os.system('detect\_mask\_video.py')

root= Tk()

root.title ("Face Mask Detection")

main\_frame= Frame(root, width=500, height=400)

main\_frame.pack()

main\_frame.place(anchor='center',relx=0.5, rely=0.5)

welcome\_text= Label(main\_frame,text="Welcome in Face Mask Detection", height=1, bg="#03A9F4", fg="#ffffff")

welcome\_text.pack()

#creating tuple

Font\_tuple = ("Comic Sans MS", 20 ,"bold")

welcome\_text.configure(font= Font\_tuple)

img = ImageTk.PhotoImage(Image.open("bg.png"))

label\_img = Label(main\_frame,image=img)

label\_img.pack()

btn\_face = Button (main\_frame, text="Face Mask Detect", command=run,height=3,width=18, bg="#006C62", fg='#fff')

btn\_face.pack(padx = 20, pady= 20)

btn\_about = Button (main\_frame, text="About US", command=about, height=2,width=12, bg="#006C62", fg='#fff')

btn\_about.pack(padx = 20, pady= 20)

btn\_exit = Button (main\_frame, text="Exit", command=root.destroy, height=2, width=8, bg="red", fg='#fff')

btn\_exit.pack(padx = 20, pady= 20)

main\_frame.pack()

main\_frame.place(anchor='center',relx=0.5, rely=0.5)

root.geometry('800x800')

root.mainloop()

**detect\_mask\_video.py**

# import the necessary packages

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input

from tensorflow.keras.preprocessing.image import img\_to\_array

from tensorflow.keras.models import load\_model

from imutils.video import VideoStream

import numpy as np

import imutils

import time

import cv2

import os

def detect\_and\_predict\_mask(frame, faceNet, maskNet):

# grab the dimensions of the frame and then construct a blob

# from it

(h, w) = frame.shape[:2]

blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),

(104.0, 177.0, 123.0))

# pass the blob through the network and obtain the face detections

faceNet.setInput(blob)

detections = faceNet.forward()

print(detections.shape)

# initialize our list of faces, their corresponding locations,

# and the list of predictions from our face mask network

faces = []

locs = []

preds = []

# loop over the detections

for i in range(0, detections.shape[2]):

# extract the confidence (i.e., probability) associated with

# the detection

confidence = detections[0, 0, i, 2]

# filter out weak detections by ensuring the confidence is

# greater than the minimum confidence

if confidence > 0.5:

# compute the (x, y)-coordinates of the bounding box for

# the object

box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])

(startX, startY, endX, endY) = box.astype("int")

# ensure the bounding boxes fall within the dimensions of

# the frame

(startX, startY) = (max(0, startX), max(0, startY))

(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

# extract the face ROI, convert it from BGR to RGB channel

# ordering, resize it to 224x224, and preprocess it

face = frame[startY:endY, startX:endX]

face = cv2.cvtColor(face, cv2.COLOR\_BGR2RGB)

face = cv2.resize(face, (224, 224))

face = img\_to\_array(face)

face = preprocess\_input(face)

# add the face and bounding boxes to their respective

# lists

faces.append(face)

locs.append((startX, startY, endX, endY))

# only make a predictions if at least one face was detected

if len(faces) > 0:

# for faster inference we'll make batch predictions on \*all\*

# faces at the same time rather than one-by-one predictions

# in the above `for` loop

faces = np.array(faces, dtype="float32")

preds = maskNet.predict(faces, batch\_size=32)

# return a 2-tuple of the face locations and their corresponding

# locations

return (locs, preds)

# load our serialized face detector model from disk

prototxtPath = r"face\_detector/deploy.prototxt"

weightsPath = r"face\_detector/res10\_300x300\_ssd\_iter\_140000.caffemodel"

faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk

maskNet = load\_model("mask\_detector.model")

# initialize the video stream

print("[INFO] starting video stream...")

vs = VideoStream(src=0).start()

# loop over the frames from the video stream

while True:

# grab the frame from the threaded video stream and resize it

# to have a maximum width of 400 pixels

frame = vs.read()

frame = imutils.resize(frame, width=400)

# detect faces in the frame and determine if they are wearing a

# face mask or not

(locs, preds) = detect\_and\_predict\_mask(frame, faceNet, maskNet)

# loop over the detected face locations and their corresponding

# locations

for (box, pred) in zip(locs, preds):

# unpack the bounding box and predictions

(startX, startY, endX, endY) = box

(mask, withoutMask) = pred

# determine the class label and color we'll use to draw

# the bounding box and text

label = "Mask" if mask > withoutMask else "No Mask"

color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

# include the probability in the label

label = "{}: {:.2f}%".format(label, max(mask, withoutMask) \* 100)

# display the label and bounding box rectangle on the output

# frame

cv2.putText(frame, label, (startX, startY - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)

cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

# show the output frame

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop

if key == ord("q"):

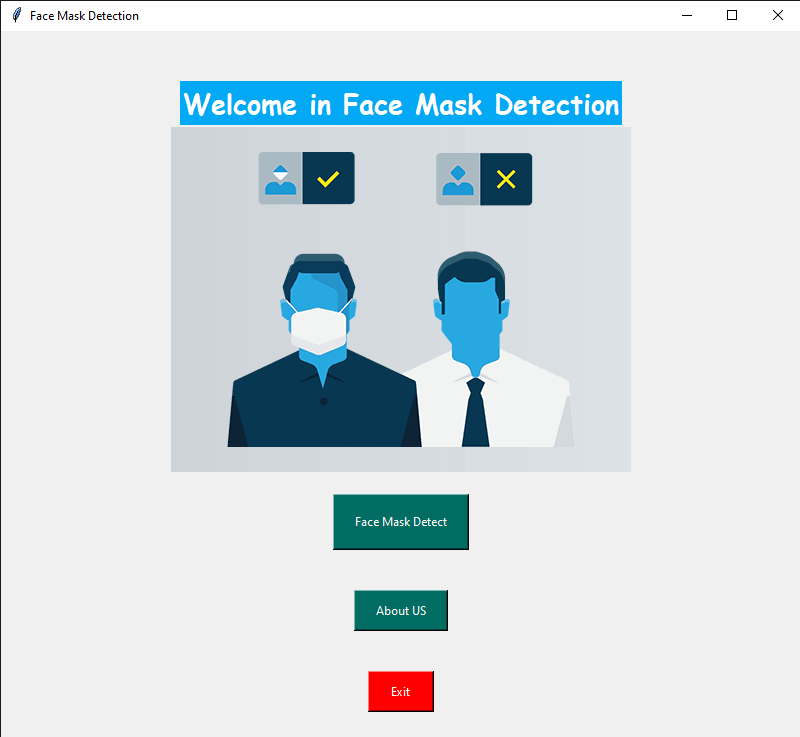
break

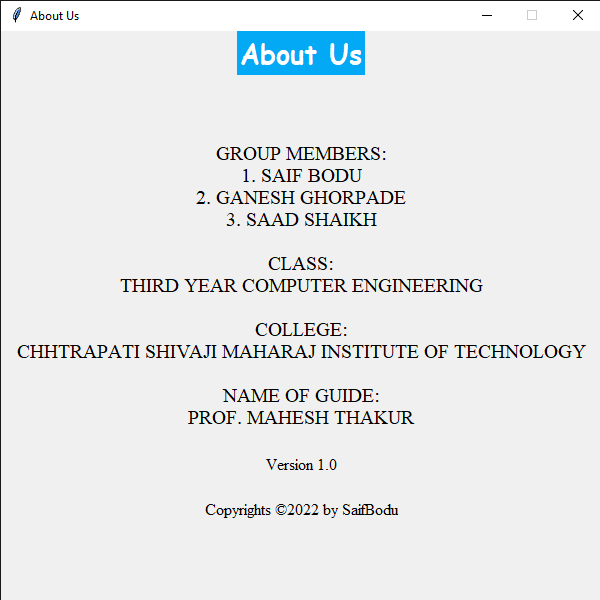
# do a bit of cleanup

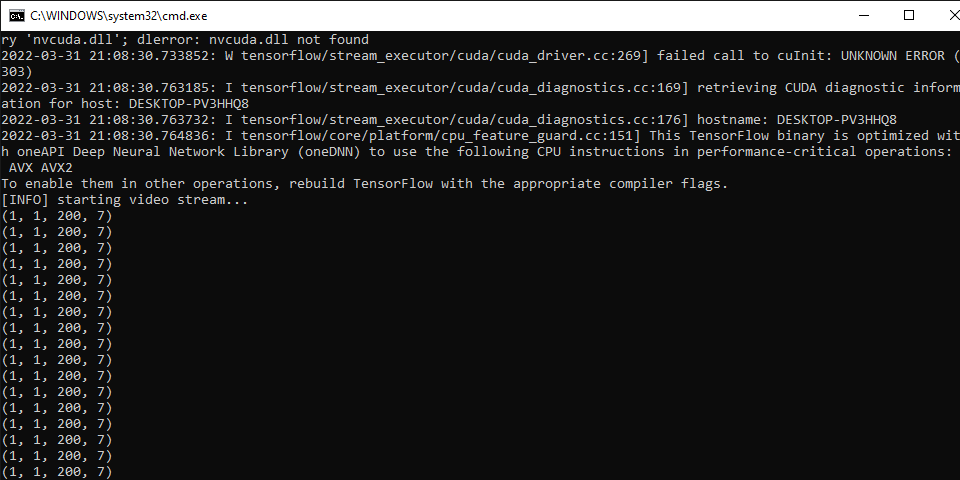
cv2.destroyAllWindows()

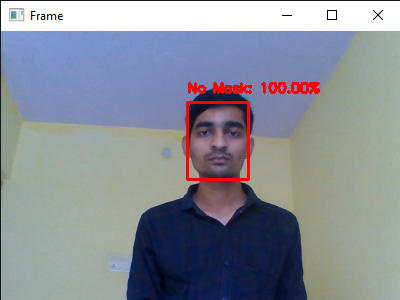
vs.stop()

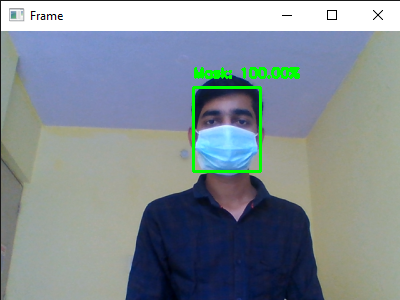
**OUTPUT**

****

****

****

****

****

**CONCLUSION**

The project was completed in stipulated time and met all the objectives. As companies are emerging with high and innovative technologies, some of the technologies should also be used for maintaining enhanced safety for the people which is turn is a benefit for the society. The models were tested real time video-streams with an optimal accuracy and as the optimization of the system is a constant process and as it is accurate solution after considering the hyper parameters and a specific model could be used as a case for the edge analytics.

**REFERENCES**

1. M. R. Bhuiyan, S. A. Khushbu and M. S. Islam, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT).
2. M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud and J. -H. Kim, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020.
3. Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification- verification,” in Advances in neural information processing systems, 2014, pp. 198hy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” 2014.
4. F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on, pp. 138–142, IEEE, 1994.
5. P. A. Rota, M. S. Oberste, S. S. Monroe, W. A. Nix, R. Campagnoli, J. P. Icenogle, S. Penaranda, B. Bankamp,K. Maher, M.-h. Chenet al., “Characterization of a novel coronavirus associated with severe acute respiratory syndrome,” science, vol. 300, no. 5624, pp. 1394–1399, 2003.
6. Z. A. Memish, A. I. Zumla, R. F. Al-Hakeem, A. A. Al Rabeeah, and G. M. Stephens, “Family cluster of middleeast respiratory syndrome coronavirus infections,”New England Journal of Medicine, vol. 368, no. 26, pp.2487–2494, 2013.
7. Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklöv, “The reproductive number of covid-19 is higher comparedto sars coronavirus,”Journal of travel medicine, 2020.
8. Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Peiet al., “Masked facerecognition dataset and application,”arXiv preprint arXiv:2003.09093, 2020.