ARDUINO BASED FACE MASK DETECTION

Project report submitted in partial fulfillment of the requirement for the degree Of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Kshitij Sharma (181017)

Sanyam Gupta (181014)

UNDER THE GUIDANCE OF

Mr. Munish Sood

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,WAKNAGHAT

May, 2022

TABLE OF CONTENTS

DECLARATION

We hereby declare that the work presented in the report entitled **"ARDUINO BASED FACE MASK DETECTION"** submitted at **Jaypee University Of Information Technology, Waknaghat, India** is an authentic record of our work carried out under the supervision of Mr. Munish Sood. We have not submitted this work elsewhere for any other degree or diploma.

Signature Kshitij Sharma(181017)

Signature Sanyam Gupta (181014)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Signature
Mr. Munish Sood
Date : 30/05/2022

Head of the Department/Project Coordinator

ACKNOWLEDGEMENT

We would like to convey our heartfelt gratitude to Mr. Munish Sood for his invaluable assistance throughout the project; without him, this would have been a far more difficult undertaking. We are grateful to Sir for his advice and regular supervision, as well as for providing important project information and for his assistance in completing the project. He was a constant source of inspiration and guidance as we worked toward the project's completion. We would like to convey our gratitude to our parents and JUIT members for their kind cooperation and encouragement in completing this project. Our gratitude and appreciation also extend to our colleagues who have assisted us in the development of the project with their skills.

Kshitij Sharma (181017)

Sanyam Gupta (181014)

LIST OF ACRONYMS & ABREVIATIONS

ML	Machine Learning
KNN	K- Nearest Neighbours
SVM	Support Vector Machines

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The spread of COVID has caused a global disaster. This virus is spread by drops of saliva that are generated from an infected person. The spread is faster and greater in public places. According to WHO, the best way to be safe in public places is to wear a proper mask. In this project report, we have introduced a way of predicting if a person is wearing a mask or not by using TensorFlow and OpenCV. To detect if a person is wearing a mask or not, a mask is detected by a bounded box selected over the face.

With the help of this project, we will be able to help the shopkeepers as well as organizations to make sure that the COVID norms are being followed and the government guidelines are being respected. We have made sure that the accuracy of this project is up to 99% and the rate of detection of many individuals at a time is also considerable.

This project is an essential when it comes to the present crisis that we are facing.

CHAPTER 1 : INTRODUCTION

COVID-19 made a huge impact on people's lives. The epidemic claimed the lives of millions of people and impacted the lives of billions more. All business establishments, education, the economy, religion, transportation, tourism, employment, entertainment, food security, and other industries were adversely affected. As of November 2020, fifty five point six million people have been infected with Coronavirus, and 1.34 million had died as a result of it, according to the WHO (World Health Organization). This is in comparison to the Black Death, which killed about 60% of Europe's population in the 14th century. It takes roughly fourteen days for the virus to mature in the body of its host and damage them after they become infected, and in the meantime, it spreads to almost everyone who comes into contact with that person. As a result, keeping track of COVID-19's spread is incredibly difficult.

COVID-19 is primarily transferred through the spit of the peoples that are diseased people coughing or sneezing. This spreads the virus to anyone who comes into direct touch (within one metre) with a person who is infected with coronavirus. As a result, the virus spreads quickly among the general public. there has been nationwide lockdowns gone, to find out weather the person is infected or not is very difficult. Face masks are an excellent way to stop a virus from spreading. Face masks have been found to be 96 percent efficient in preventing the spread of infection. Governments all across the world have imposed strict restrictions mandating everyone to wear masks when they leave the house but it is seen that some homosapians have hot been wearing the protection shields may not wear masks, and determining whether or not everyone is wearing a face cover is challenging. In these cases, computer vision will come in helpful.

There are no reliable applications for detecting whether or not someone is wearing a mask. This project uses machine learning classification using OpenCV and TensorFlow to detect facemasks on humans.

Figure 1.1. : Person with mask

To prevent diseases like these we have to wear a face cover that can be a mask, such as COVID-19. The public should know whether to wear the mask for source control or COVID-19 aversion. The use of masks has the potential to reduce vulnerability to danger from a noxious individual during the "pre-symptomatic" stage, as well as stigmatize individuals who use masks to prevent the spread of virus. The World Health Organization emphasizes the use of medical masks and respirators for health care assistants. Face mask detection entails identifying the location of a person's face and then assessing whether or not they are wearing a mask. The issue is with general object detection, which is used to recognise various types of things. The process of recognising a certain collection of items, such as faces, is known as face identification. It can be used for a variety of things, such as autonomous driving, education, and surveillance.

CHAPTER 2: BACKGROUND

2.1 : COVID-19

The coronavirus has infected over 266 million people globally, killing over 5 million, according to the World Health Organization (WHO). SARS (Severe Acute Respiratory Syndrome) and MERS (Middle East Respiratory Syndrome) are two dangerous respiratory illnesses that have spread over the Middle East (MERS). In recent years, these illnesses have become substantially more common As a result, people should be aware about their well-being and breathing issues. Every country's government considers public health to be a top priority.. The WHO also stated that masking faces can assist to prevent the spread of coronavirus, and the government has made it mandatory for everyone to wear face masks when they go to any public place. People with breathing issues, on the other hand, should definitely cover their faces. As a result, several service providers and supermarkets demand clients to use their services while wearing masks. This is why a face mask detecting system is required to assist people, although just a few studies have been conducted on the subject. Face mask detection is a technology that detects whether or not a person is wearing a mask.. It's similar to an object detection system, which detects a certain type of thing. We hope that through constructing this system, we would be able to help in making sure people's well-being in public places. This method can be used in a variety of places, including big markets and malls, schools, colleges, and stations, among others. In this research, we present a face mask detection system that can determine whether or not a person is wearing a mask. It's a two-phase technique in which we first train our system before using it to recognise faces. We used Keras Tensorflow and OpenCV to improve our results, and we trained our model with a big dataset of with and without mask faces. We chose this architecture because deploying our face mask detector to embedded devices could minimise the cost of manufacturing such face mask detection systems.

2.2 : PROPOSED METHODOLOGY

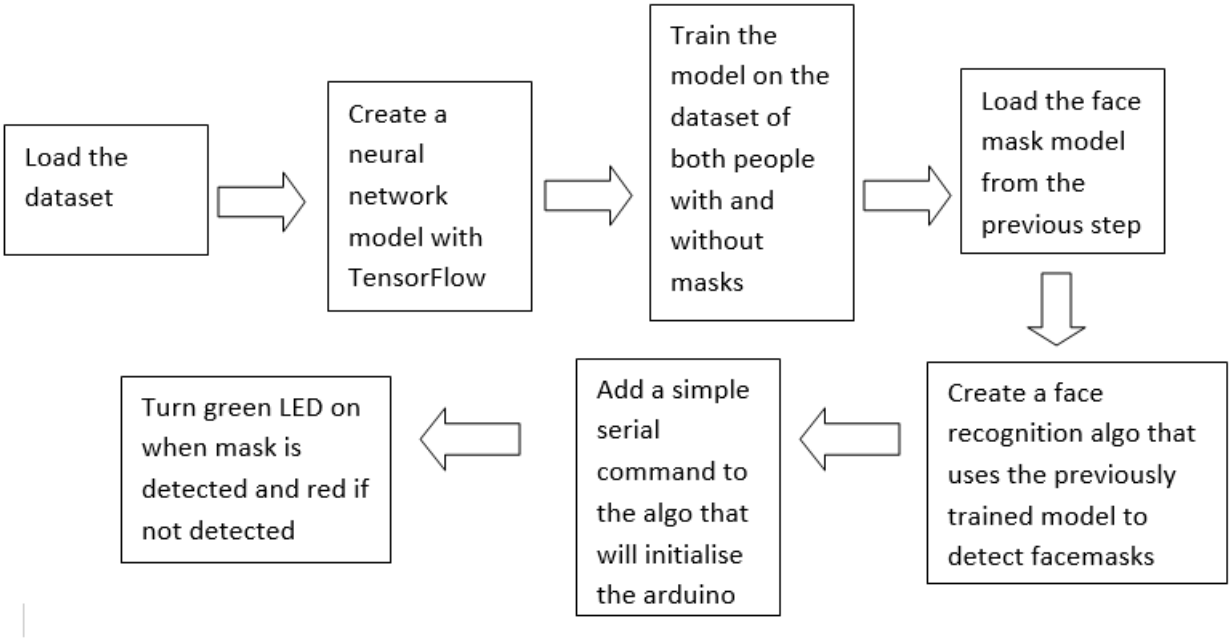


Figure 2.2: Block diagram of project

2.3 : TECHNOLOGY TO BE USED

Microsoft Visual Studio Code

Visual Studio Code is a cross-platform integrated development environment (IDE) for the Windows, Linux, and macOS that was developed by Microsoft. There are a lot of the features some of them are debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, and options, as well as install extensions for additional functionality.

Python Libraries

The Python Library is a reusable programming component that you can use in your own programs or tasks. Python libraries, unlike those written in languages like C++ or C, are not bound to a specific context. The phrase 'library' refers to a collection of simple modules.

Table2.1 : Python Libraries Used

S.NO.	PYTHON LIBRARIES	USES
1.	TensorFlow	It can be used for a variety of applications, but it focuses on deep neural network training and inference.
2.	Keras	Python library ..
4.	Numpy	Numpy .
5.	OpenCV	OpenCV is a library of programming functions mainly aimed at real-time computer vision.
6.	Scipy	SciPy is a free and open-source Python library used for scientific computing and technical computing.
7.	Sklearn	Scikit learn library of python.

Arduino

Arduino is a company, an effort, and a user community that creates single-board microcontrollers and microcontroller kits for building digital devices with open-source hardware and software.

CHAPTER 3 : ARDUINO BASED FACE MASK DETECTION

3.1 : TENSORFLOW, KERAS, OPENCV

TensorFlow

Google researchers created TensorFlow, an open source framework for performing machine learning, deep learning, and other statistical and predictive analytics workloads. Its goal, like those of similar platforms, is to make it easier for users like data scientists, statisticians, and predictive modelers to create and implement advanced analytics applications. Google is also one of the platforms that uses this kind of application in their applications . it also uses auto generated emails and also recognise them as well.

`

Keras

Deep neural networks , but they've proven challenging to use for developers who aren't familiar with machine learning. The ideas appear similar at first glance but differ when studied more closely. Keras (a Python-based deep learning API that sits on top of the TensorFlow machine learning platform) is one of the high-level neural network APIs It was made with the intention of facilitating speedy experimentation. When conducting research, it's critical to be able to move rapidly from concept to conclusion. Many implementations of standard neural-network building blocks like layers, objectives, activation functions, optimizers, and a slew of other tools are included in Keras to make working with picture and text data easier while also reducing the amount of coding required to write deep neural network code. The source code is hosted on GitHub, and there's a Slack channel and a GitHub issues page for community support.

OpenCV

OpenCV (Open source computer vision library) is a programming library that focuses on real-time computer vision. The library is platform agnostic. Open cv was one of the first version of computer that was created These algorithms can be used to recognise and detect faces, identify items, and track moving objects, a it has both all the libraries as well as newest technologies among other things.

3.2 : DEEPLEARNING

Deep learning is the type of machine learning that works the similar way as our brain works is helps in creating pattern and pictures similar our brain . We'll use TensorFlow to build a neural network model that will be trained on a dataset of persons wearing and not wearing facemasks.

However, if we run my code without modifying the Model settings, we can ensure a 98 percent confidence level.

3.3 : PHASES

We have distributed this project in to two parts, each with its own set of sub-steps, in order to train a custom face mask detector:

● Training: the model in this part of the project we are gathering the images with masks and images without masks and making our model differentiate between them.

● Deployment: After training the face mask detector, we loaded with the program that finds out whether you are wearing mask, performed face detection, and then classified each face as having or not having a mask.

3.4 : DATASET

Karem Ben Chika created the dataset that we used in the research. There are 3833 photos in this collection, divided into two categories:

● Wearing a mask: 1915 images

● not wearing a mask: 1918 images

The motive is to develop a bespoke that can find out whether or not someone is having their face covered up with a masks Karem devised an interesting technique to create this dataset :

● Taking normal images of faces

● After that, you'll need to make a bespoke computer vision. Using a Python script, I was able put a face cover to them, resulting in an artificial (but still useful) dataset. To create the discrption of persons covering their face with a masks, facial expressions are added to the face. Face landmarks help us to infer the placement of facial elements such as the eyes, brows, nose, mouth, and jawline automatically.

3.5 : IMPLEMENTATION

We'll use TensorFlow to build a neural network model that will be trained on a dataset of persons wearing and not wearing facemasks.

However, if we run my code without modifying the Model settings, we can ensure a 98 percent confidence level.

Then, using the trained model from the previous phase, we'll develop a face recognition algorithm that can recognise facemasks on people's faces.

We used a pre-trained model that can be used in this stage without having to go through each step individually.

Finally, we'll add a simple Serial Command to the facemask detection method, instructing the Arduino to turn on or off LEDs depending on the detection condition.

3.6 : CNN

CNN convolutional neural network it is a artificial neural network which is used for the purpose of pixel processing of a image which helps us in differentiate between its

CNNs are image processing, artificial intelligence (AI) systems that employ deep learning to do both generative and descriptive tasks, frequently with the assistance of machine vision, such as picture and video identification, recommender systems, and natural language processing (NLP).A neural network is a hardware and/or software system that mimics the behaviour of brain neurons. Al the other neural networks were not designed to process image but the CNN is can be used in image recognition.

However, if we run my code without modifying the Model settings, we can ensure a 98 percent confidence level.

Then, using the trained model from the previous phase, we'll develop a face recognition algorithm that can recognise facemasks on people's faces.

We used a pre-trained model that can be used in this stage without having to go through each step individually.

A deep neural network (DNN), often known as a deep net, is a neural network with multiple layers, usually at least two. Deep nets process data in sophisticated ways using advanced math modelling.

Deep neural networks are a type of machine learning approach that uses layers of neural networks to generate deeper and wider structures. Deep networks have demonstrated discriminative and representation learning abilities across a wide range of applications in recent years.

CHAPTER 4: RESULTS AND DISCUSSION

By following the above mentioned methodology, we were able to achieve our desired goal of creating the project and have achieved the following results which are up to the standards predicted prior to the implementation of the project.

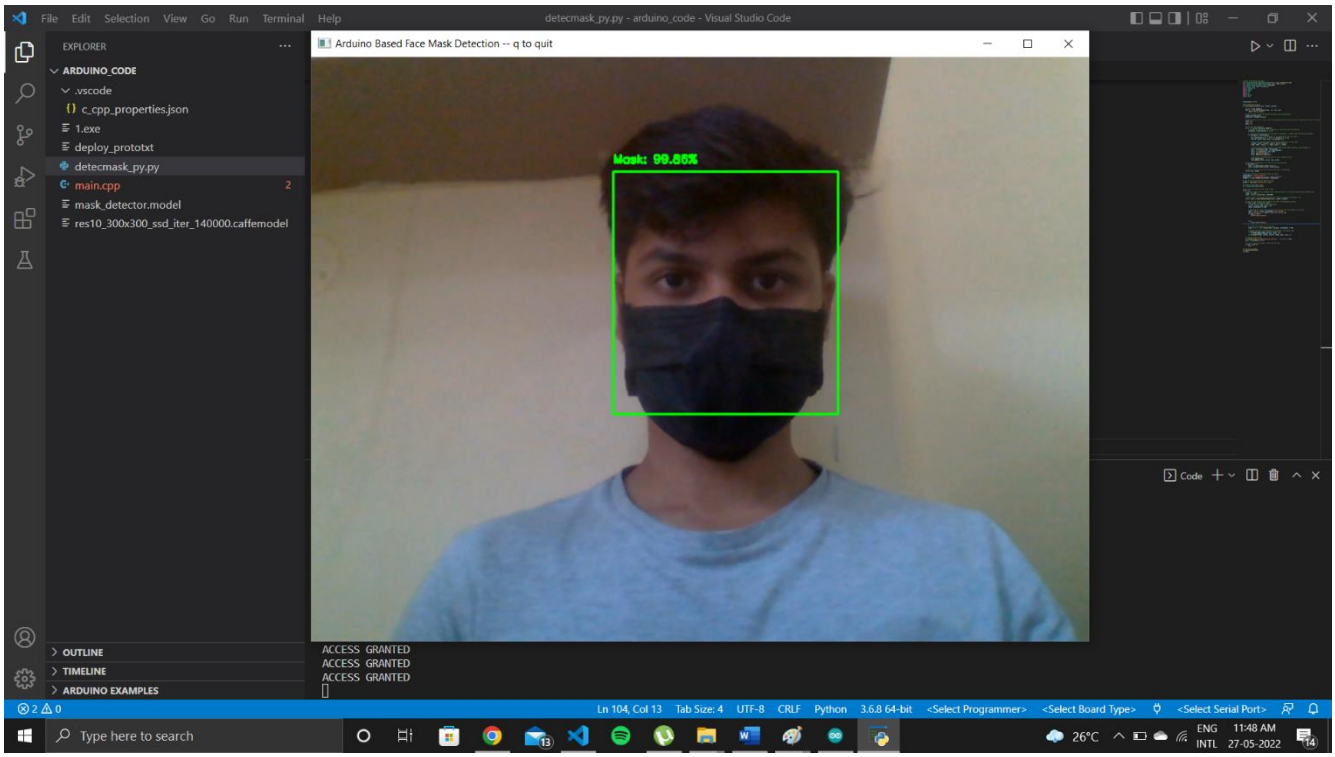


Figure 4.1. : With mask

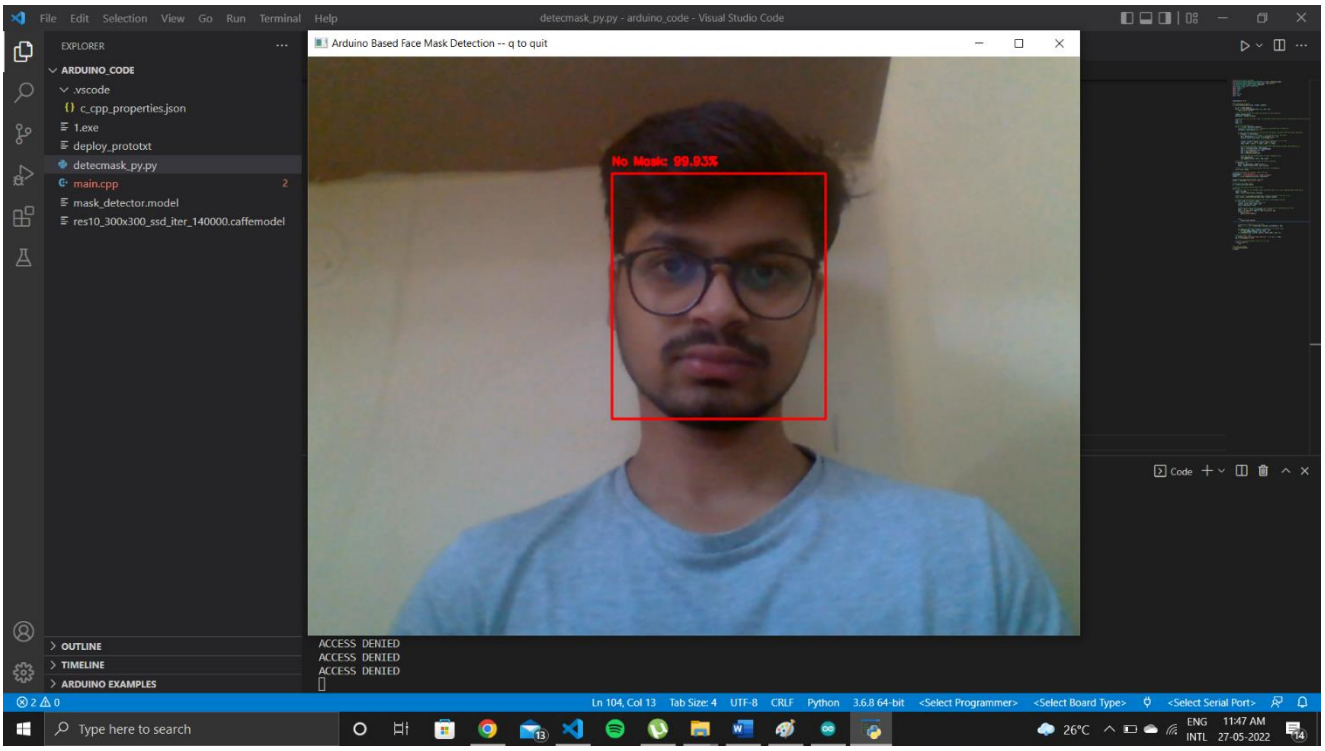


Figure 4.2. : Without mask

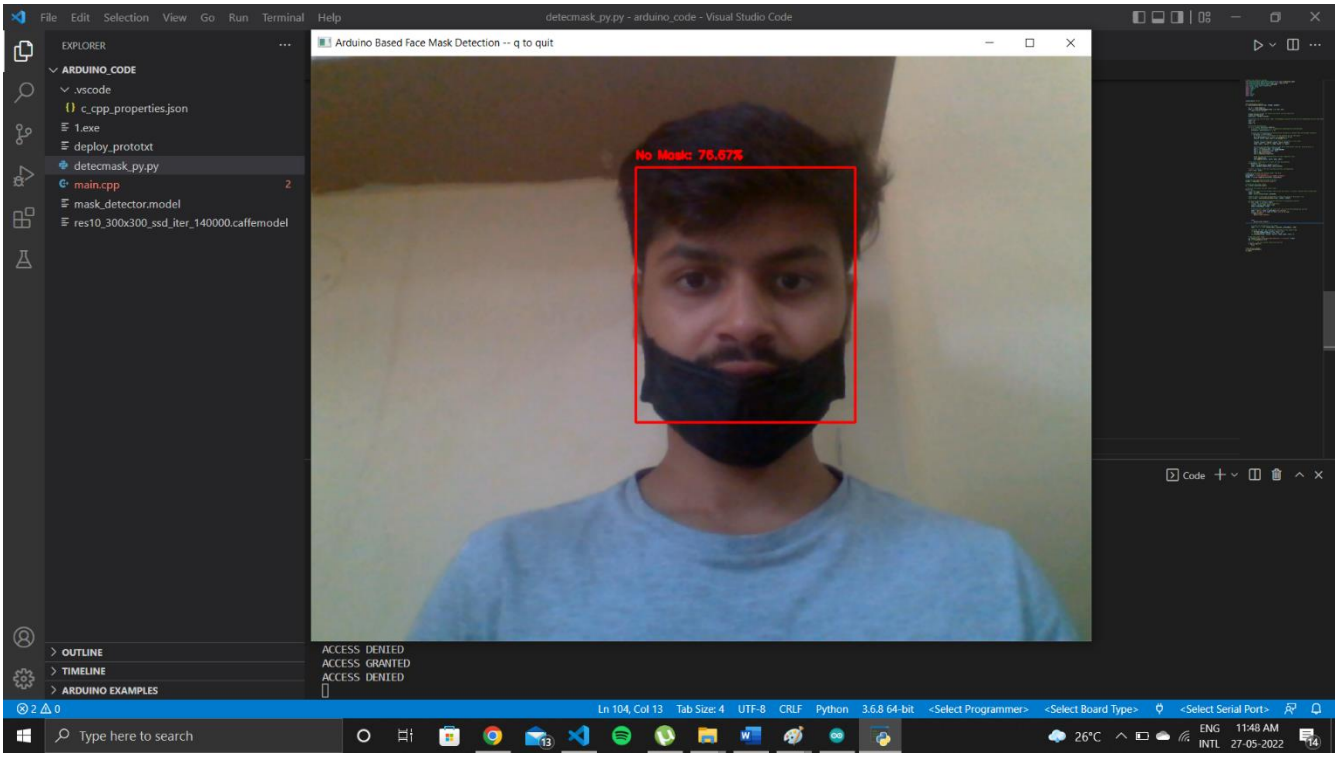


Figure 4.3. : With partial mask

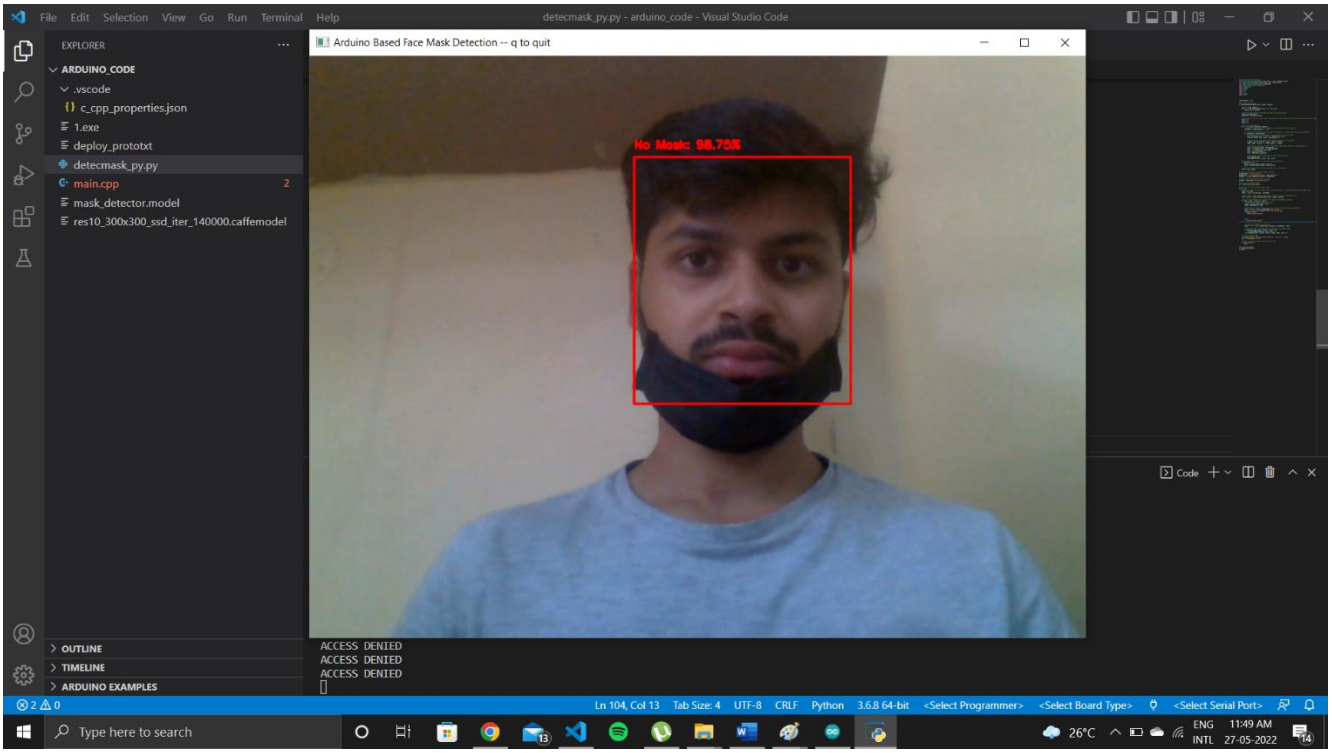


Figure 4.4. : With zyada partial mask

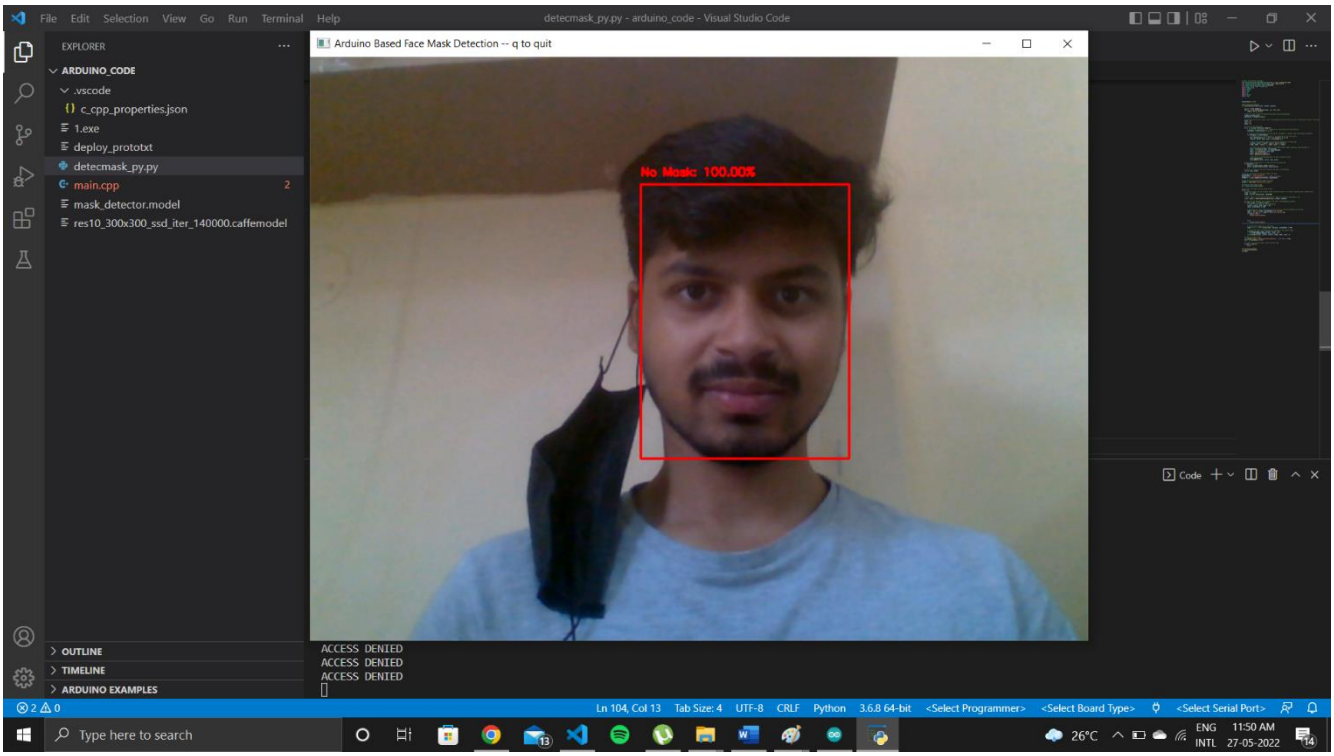


Figure 4.5. : Uncleji style mask

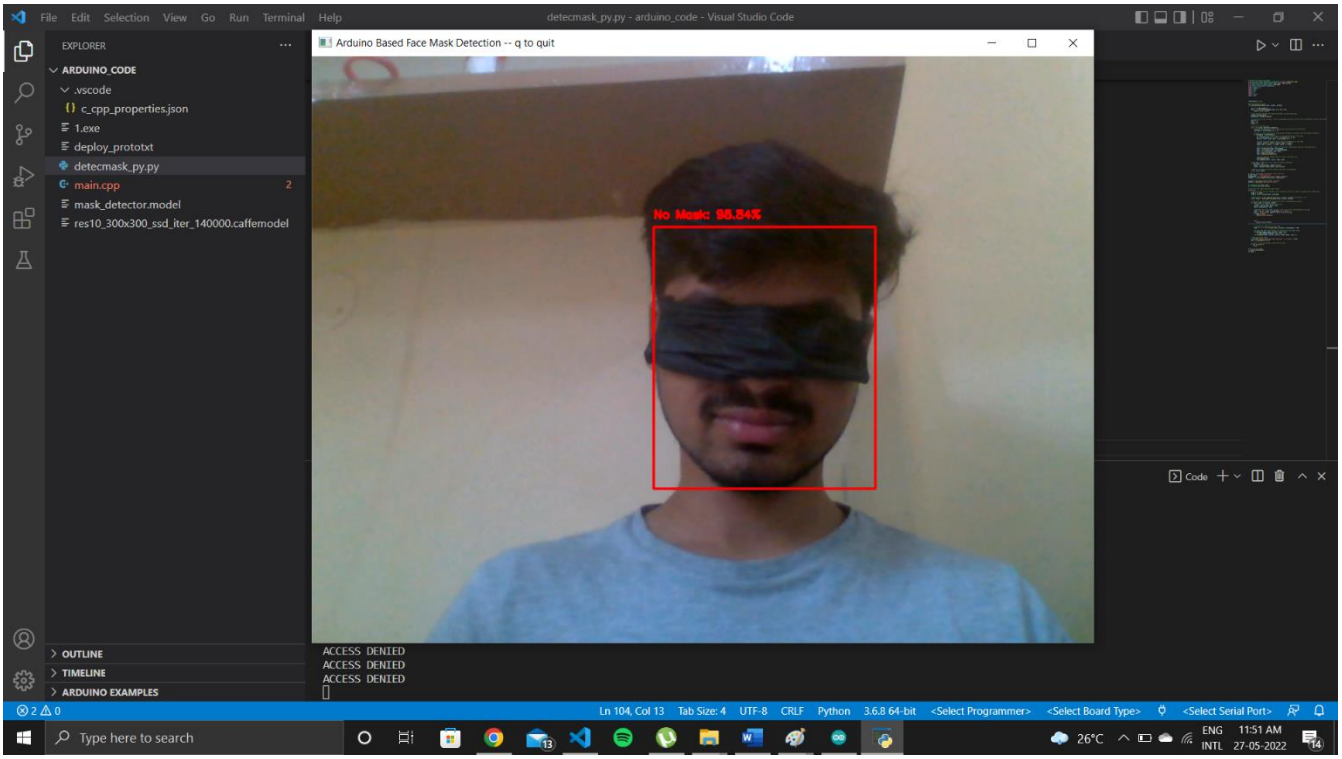
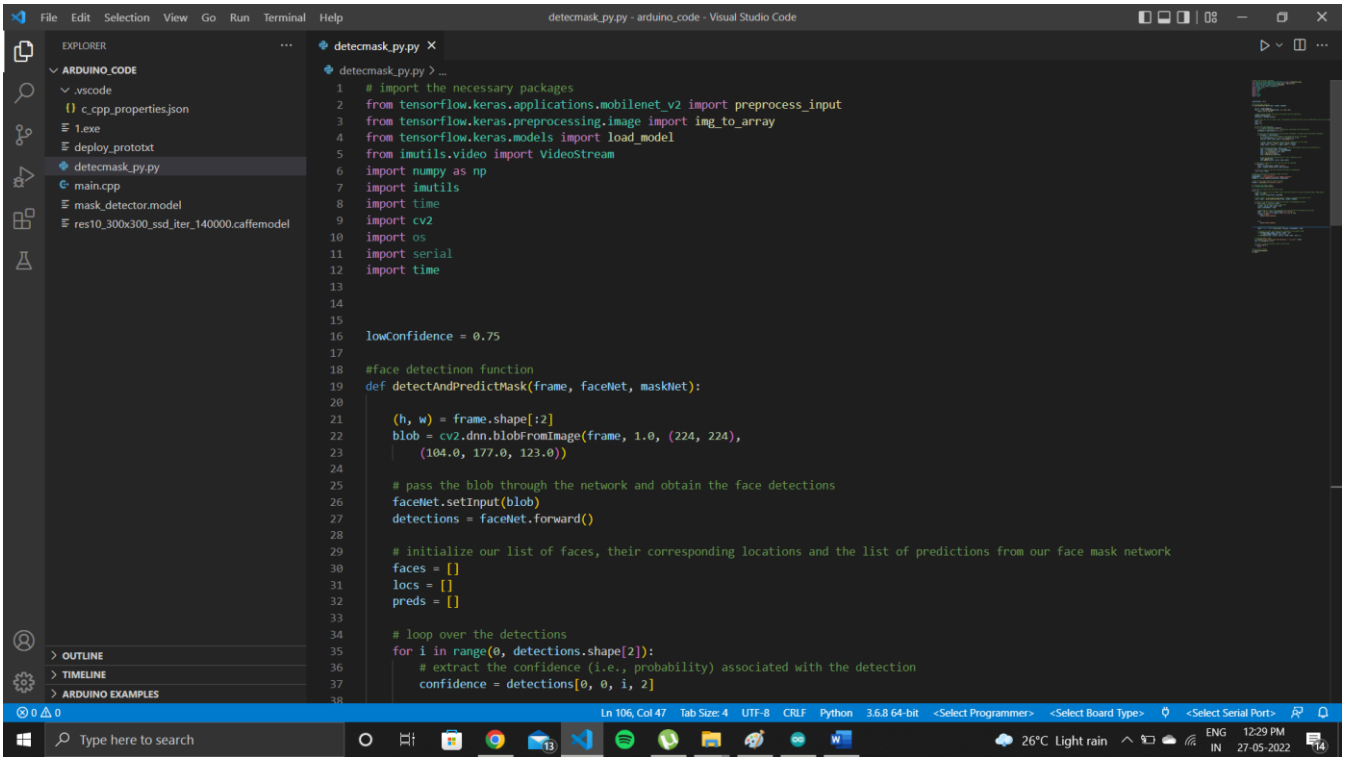
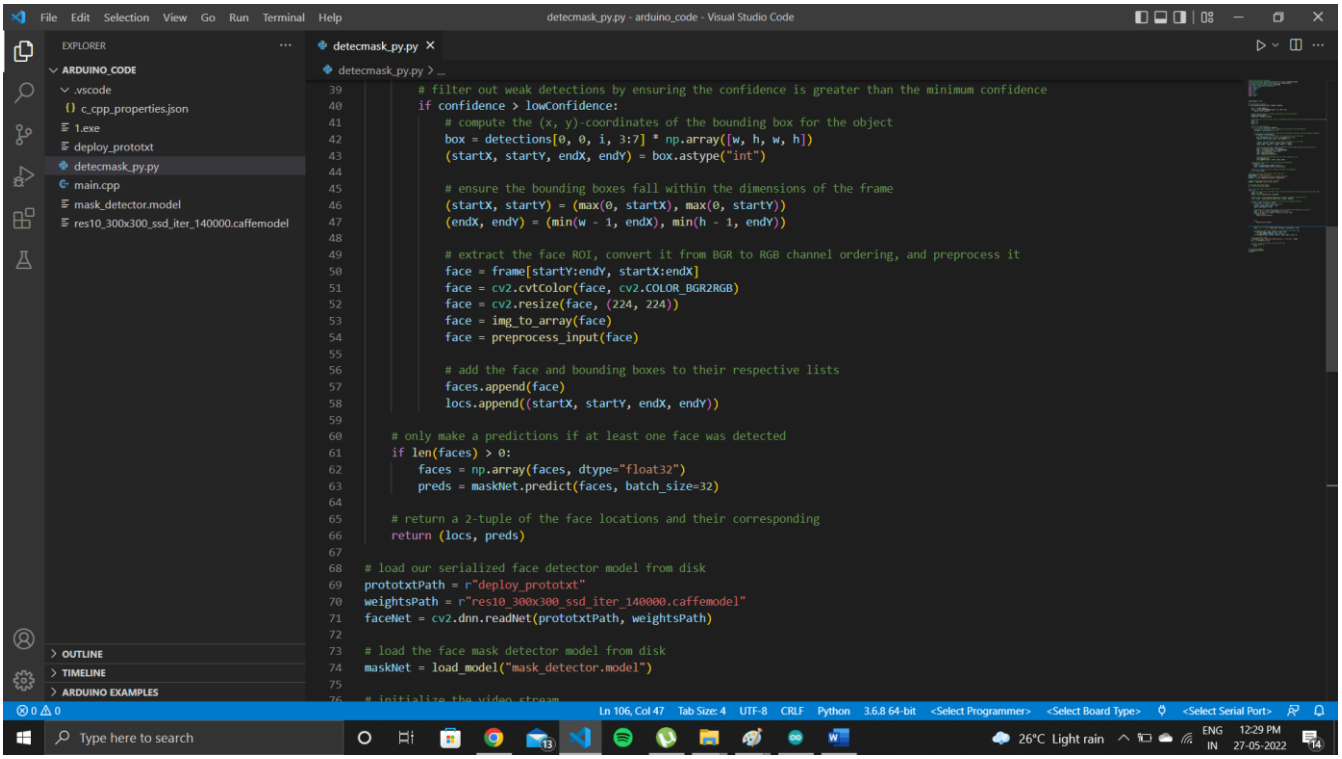
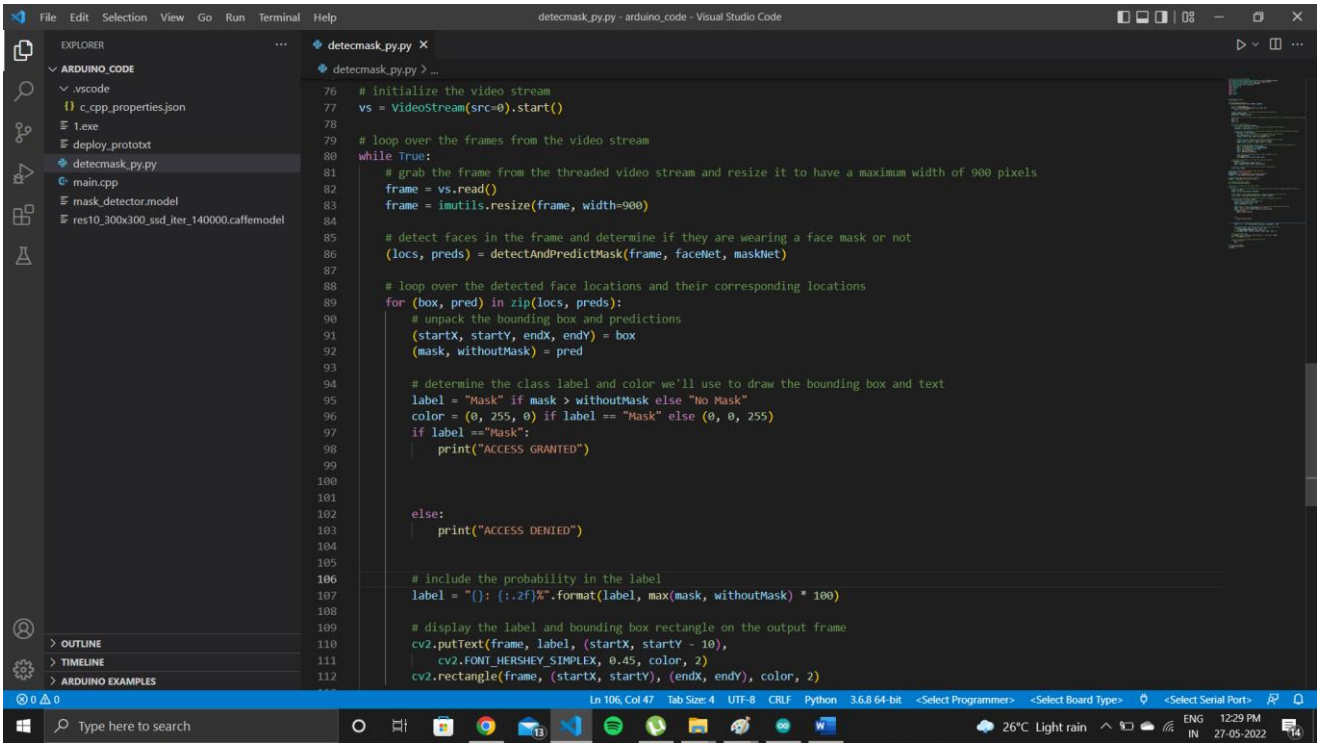
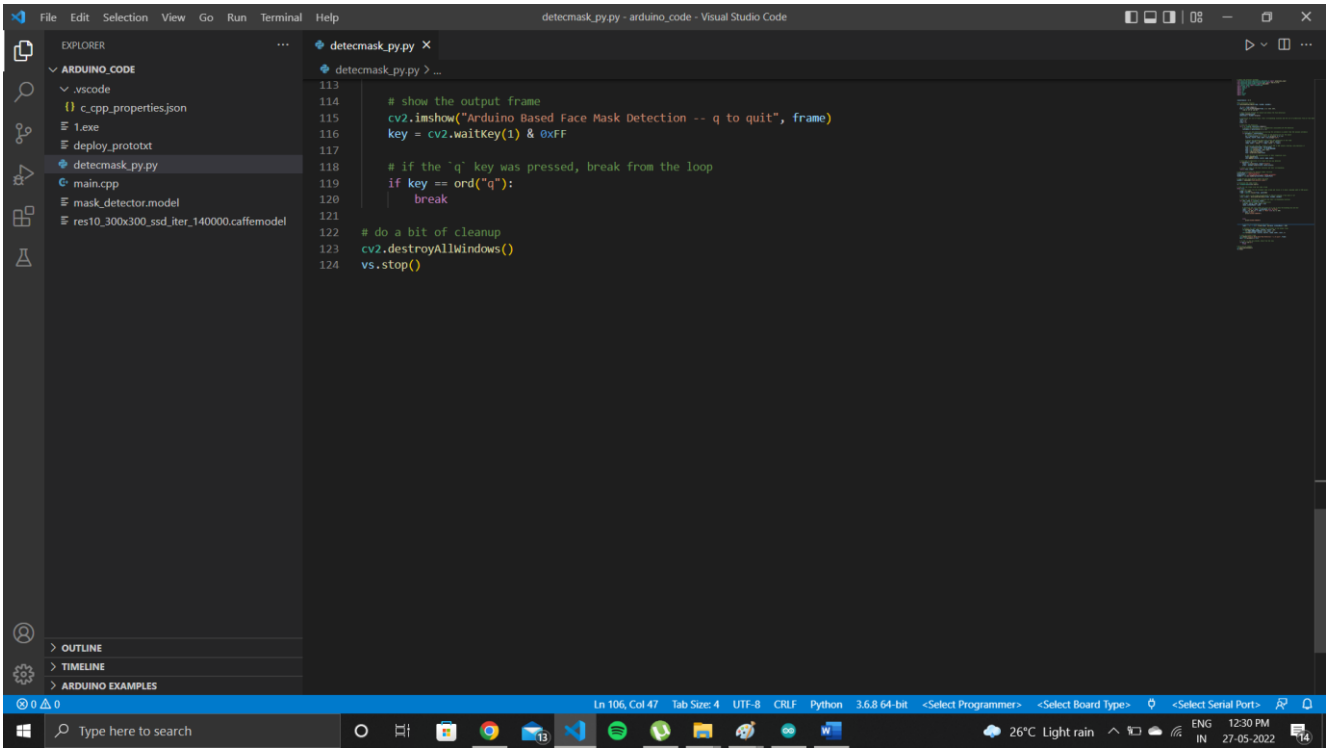


Figure 4.6. : Blind mask man


File Edit Selection View Go Run Terminal Help
detecmask_py.py - arduino_code - Visual Studio Code
EXPLORER
ARDUINO_CODE
.vscode
c_cpp_properties.json
1.exe
deploy_prototxt
detecmask_py.py
main.cpp
mask_detector.model
res10_300x300_ssd_iter_140000.caffemodel
OUTLINE
TIMELINE
ARDUINO EXAMPLES
detecmask_py.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
import serial
import time
lowConfidence = 0.75
#face detectinon function
def detectAndPredictMask(frame, faceNet, maskNet):
(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()
# 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]
Ln 106, Col 47 Tab Size: 4 UTF-8 CRLF Python 3.6.8 64-bit <Select Programmer> <Select Board Type> <Select Serial Port>
Type here to search
26°C Light rain
ENG IN
12:29 PM 27-05-2022


File Edit Selection View Go Run Terminal Help
detecmask_py.py - arduino_code - Visual Studio Code
EXPLORER
ARDUINO_CODE
.vscode
c_cpp_properties.json
1.exe
deploy_prototxt
detecmask_py.py
main.cpp
mask_detector.model
res10_300x300_ssd_iter_140000.caffemodel
OUTLINE
TIMELINE
ARDUINO EXAMPLES
detecmask_py.py
# filter out weak detections by ensuring the confidence is greater than the minimum confidence
if confidence > lowConfidence:
# 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, 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:
faces = np.array(faces, dtype="float32")
preds = maskNet.predict(faces, batch_size=32)
# return a 2-tuple of the face locations and their corresponding
return (locs, preds)
# load our serialized face detector model from disk
prototxtPath = r"deploy_prototxt"
weightsPath = r"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")
Ln 106, Col 47 Tab Size: 4 UTF-8 CRLF Python 3.6.8 64-bit <Select Programmer> <Select Board Type> <Select Serial Port>
Type here to search
26°C Light rain
ENG IN
12:29 PM 27-05-2022


File Edit Selection View Go Run Terminal Help
detecmask_py.py - arduino_code - Visual Studio Code
EXPLORER
ARDUINO_CODE
.vscode
c_cpp_properties.json
1.exe
deploy_prototxt
detecmask_py.py
main.cpp
mask_detector.model
res10_300x300_ssd_iter_140000.caffemodel
OUTLINE
TIMELINE
ARDUINO EXAMPLES
detecmask_py.py
detecmask_py.py > ...
# initialize the 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 900 pixels
frame = vs.read()
frame = imutils.resize(frame, width=900)
# detect faces in the frame and determine if they are wearing a face mask or not
(locs, preds) = detectAndPredictMask(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)
if label =="Mask":
print("ACCESS GRANTED")
else:
print("ACCESS DENIED")
# 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)
Ln 106, Col 47 Tab Size: 4 UTF-8 CRLF Python 3.6.8 64-bit <Select Programmer> <Select Board Type> <Select Serial Port>
Type here to search
26°C Light rain
ENG IN
12:29 PM
27-05-2022


File Edit Selection View Go Run Terminal Help
detecmask_py.py - arduino_code - Visual Studio Code
EXPLORER
ARDUINO_CODE
.vscode
c_cpp_properties.json
1.exe
deploy_prototxt
detecmask_py.py
main.cpp
mask_detector.model
res10_300x300_ssd_iter_140000.caffemodel
OUTLINE
TIMELINE
ARDUINO EXAMPLES
detecmask_py.py
detecmask_py.py > ...
# show the output frame
cv2.imshow("Arduino Based Face Mask Detection -- q to quit", 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()
Ln 106, Col 47 Tab Size: 4 UTF-8 CRLF Python 3.6.8 64-bit <Select Programmer> <Select Board Type> <Select Serial Port>
Type here to search
26°C Light rain
ENG IN
12:30 PM
27-05-2022

REFERENCES

1. Rajat Sachdeva, Sonam, Harshita Sharma (2020), “Face Mask Detection System”. International Journal of Scientific and Engineering Research, Volume 11, Issue 12. ISSN 2229-5518.
2. Arjya Das, Mohammed Wasif Ansari, Rohini Basak (2020). “Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV”. IEEE 17th India Council International Conference (INDICON).
3. Harish Adusumalli, D. Kalyani, R.Krishna Sri, M. Pratapteja, P V R D Prasada Rao (2021), “Face Mask Detection Using OpenCV ”. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks(ICICV).
4. Daniel Matthias, Chidozie Managwu, “COVID-19 Project”.