

Face Mask Detection System

WHAT

1. It is Computer Vision Application which can detect whether a person is wearing a mask or not and then store their data who are not wearing the mask.
2. This Application can get data from various sources such as IP Camera, Webcam, online video.
3. This is a web application which can be accessed over a LAN.
4. The IP Camera is a wireless surveillance camera which can send footage to Wi-Fi network to which it is connected then if the Computer which contain the application is also connected with the same Wi-Fi network, then we will be able work on that footage data into our project and detect mask.

WHY

1. This Application can be used in many places such as hospitals, Research labs, Polluted Area, Air borne pandemic.
2. This kind of project is shows the machine learning knowledge, programming skills and web development skills which can be very useful for resume.
3. This kind of project shows you can solve the real world problem with your knowledge.

HOW

1)Backend :

Steps	Library Used
Connection with IP Camera	OpenCV
Face Detection	OpenCV
Mask Detection	Keras, numpy
Save Data(Person not wearing mask)	OpenCV

Frontend : Web Application -> Streamlit

Make a new Folder by the name of Facemask name

Go to program/python/python 310

Open cmd

And write **python -m venv your project url/path location**

Come to folder facemask under script folder ->open cmd and **activate** it

install -> opencv

pip install opencv -python

Connection with IP Camera

Backend.py file

#Getting a data from life-streaming IP Camera or WebCam

```
import cv2

vid=cv2.VideoCapture(0)

while(vid.isOpened()):

    flag,frame=vid.read()

    if(flag):

        cv2.namedWindow("Window",cv2.WINDOW_NORMAL)

        cv2.imshow("Facemask window",frame)

        key=cv2.waitKey(20)

        if(key==ord('x')):

            break

    else:

        break

vid.release()

cv2.destroyAllWindows()
```

Note: For using IP Camera we just give url in VideoCapture function

```
cv2.VideoCapture("IP CAMERA URL")
```

Face Detection

- 1) Download one video of facemask from www.pexels.com
- 2) Using Face.xml(It represent data of faces like geometric of face and different different info about face) for detecting face.

Code for Detect Face and Save The Face Into A Folder Of Your System

```
import cv2

facemodel=cv2.CascadeClassifier("face.xml")

vid=cv2.VideoCapture("http://192.168.0.124:8080/video")

i=1

while(vid.isOpened()):
```

```

flag,frame=vid.read()

if(flag):

    faces=facemodel.detectMultiScale(frame)

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

        face_img=frame[y:y+w,x:x+l]

        path="Data/"+str(i)+".jpg"

        i=i+1

        cv2.imwrite(path,face_img)

        cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,0),3)

cv2.namedWindow("Facemask window",cv2.WINDOW_NORMAL)

cv2.imshow("Facemask window",frame)

key=cv2.waitKey(20)

if(key==ord('x')):

    break

else:

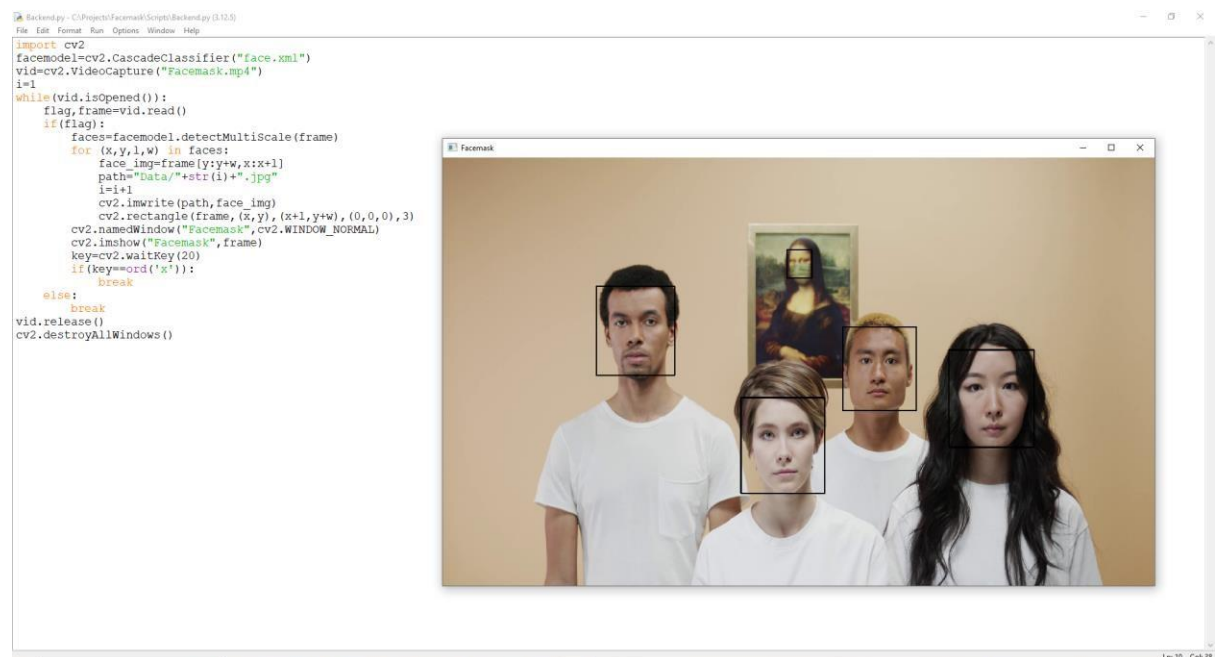
    break

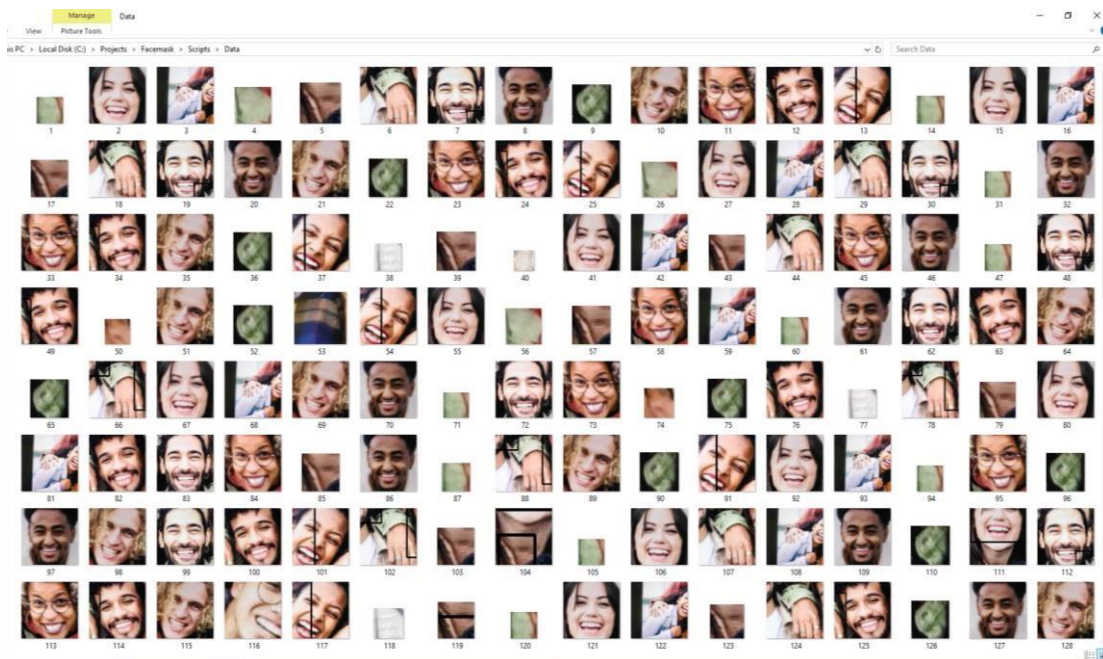
vid.release()

cv2.destroyAllWindows()

```

output :





Install package

- pip install pillow
- pip install scipy
- pip install tensorflow

We are Using CNN Concept to train Data and create model

We have 2 folders train and test data which include with and without mask images of persons.

train.py file

```
from keras.layers import Conv2D,MaxPooling2D,Flatten,Dense
```

```
from keras.models import Sequential
```

```
from tensorflow.keras.optimizers import Adam
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

#Define the model

```
mymodel=Sequential()

mymodel.add(Conv2D(32,(3,3),activation='relu',input_shape=(150,150,3)))

mymodel.add(MaxPooling2D())

mymodel.add(Conv2D(32,(3,3),activation='relu'))

mymodel.add(MaxPooling2D())

mymodel.add(Conv2D(32,(3,3),activation='relu'))

mymodel.add(MaxPooling2D())

mymodel.add(Flatten())

mymodel.add(Dense(100,activation='relu'))

mymodel.add(Dense(1,activation='sigmoid'))

mymodel.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['accuracy'])
```

#Define the Data

```
train=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)

test=ImageDataGenerator(rescale=1./255)

train_img=train.flow_from_directory('train', target_size=(150,150), batch_size=16, class_mode='binary')

test_img=test.flow_from_directory('test', target_size=(150,150), batch_size=16, class_mode='binary')
```

#Train and Test the model

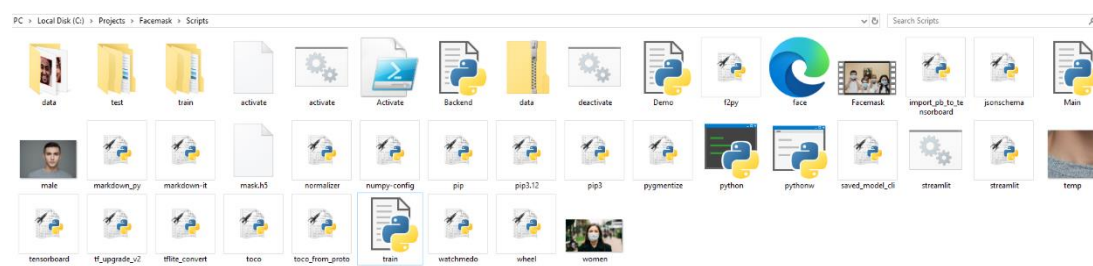
```
mask_model=mymodel.fit(train_img, epochs=10, validation_data=test_img)
```

#Save the model in my Directory

```
mymodel.save('mask.h5', mask_model)
```

output:-

It will create model with a name of mask.h5 extension



To Read And Image And Detect Mask

In cmd if we provide image it classify into 1 and 0 form

So,1 means person not wearing the mask and complete face is available

0 means person wearing the mask and incomplete face is available

Backend.py

```
import cv2

from tensorflow.keras.models import load_model
from keras.utils import load_img, img_to_array
import numpy as np

#To read an image detect mask and show it
facemodel=cv2.CascadeClassifier("face.xml")
maskmodel=load_model('mask.h5')
vid=cv2.VideoCapture("facemask.mp4")

i=1
while(vid.isOpened()):
    flag,frame=vid.read()
    if(flag):
        faces=facemodel.detectMultiScale(frame)
        for (x,y,l,w) in faces:
            crop_face1=frame[y:y+w,x:x+l]
            cv2.imwrite('temp.jpg',crop_face1)
            crop_face=load_img('temp.jpg',target_size=(150,150,3))
            crop_face=img_to_array(crop_face)
            crop_face=np.expand_dims(crop_face,axis=0)
            pred=maskmodel.predict(crop_face)[0][0]
            if pred==1:
                cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,255),4)
                path="data/"+str(i)+".jpg"
                cv2.imwrite(path,crop_face1)
                i=i+1
            else:
```

```
cv2.rectangle(frame,(x,y),(x+l,y+w),(0,255,0),4)
```

```
cv2.namedWindow("Facemask",cv2.WINDOW_NORMAL)
```

```
cv2.imshow("Facemask",frame)
```

```
key=cv2.waitKey(15)
```

```
if(key==ord('x')):
```

break

```
else:
```

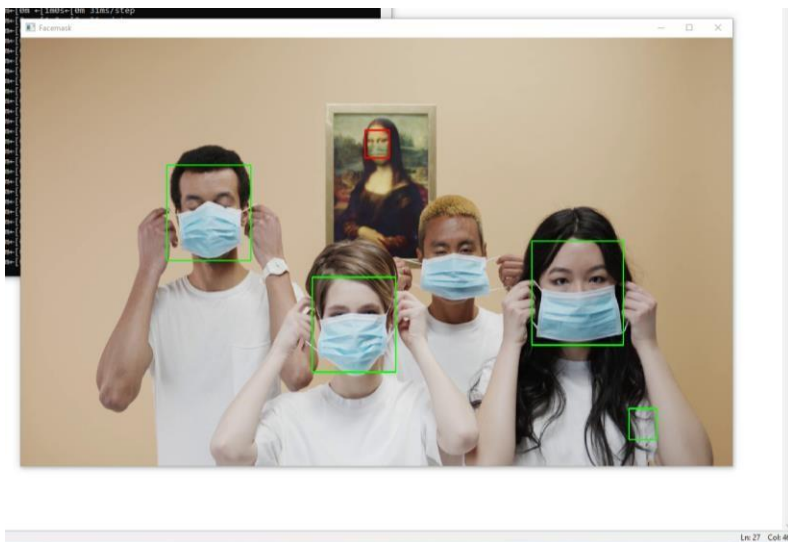
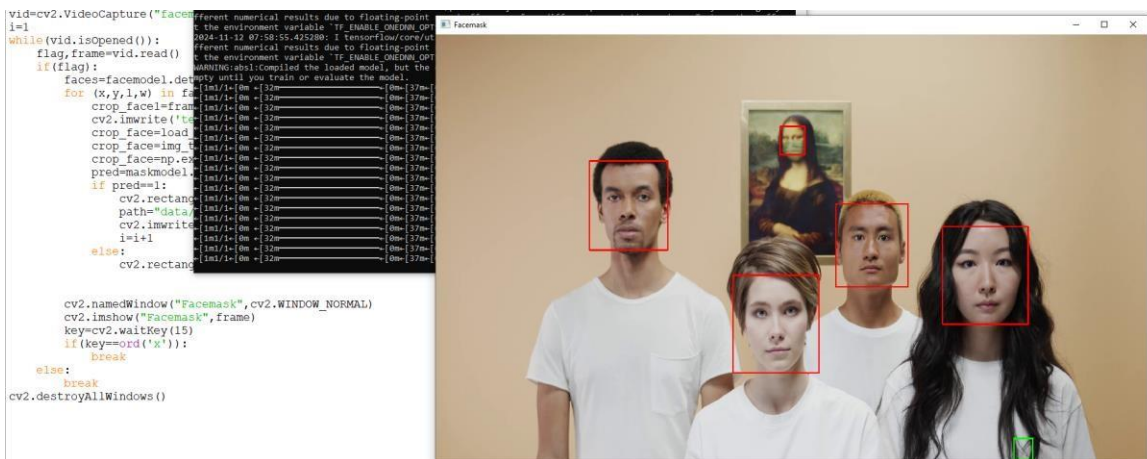
break

cv2.destroyAllWindows()

Output:

Note: Red Rectangle: Person not wearing a mask.

Green Rectangle: Peron is wearing a mask.



Person who are not wear mask save in data folder



Creating Web Application For Face Mask Detection Using Streamlit

Install package

pip install streamlit

Main.py

```
import streamlit as st
```

```
from keras.models import load_model
```

```
from keras.utils import load_img, img_to_array
```

```
import numpy as np
```

```
import tempfile
```

```
import cv2
```

```
facemodel=cv2.CascadeClassifier("face.xml")
```

```
st.title("Face Mask Detection System")
```

```
choice=st.sidebar.selectbox("MY MENU",("HOME","IMAGE","VIDEO","CAMERA"))
```

```
if(choice=='HOME'):
```

```
    st.header("WELCOME")
```

```
elif(choice=="IMAGE"):
```

```
    file=st.file_uploader("Upload Image")
```

```
    if file:
```



```

b=file.getvalue()
d=np.frombuffer(b,np.uint8)
img=cv2.imdecode(d,cv2.IMREAD_COLOR)
face=facemodel.detectMultiScale(img)
maskmodel=load_model('mask.h5')
for(x,y,l,w) in face:
    crop_face=img[y:y+w,x:x+l]
    cv2.imwrite('temp.jpg',crop_face)
    crop_face=load_img('temp.jpg',target_size=(150,150,3))
    crop_face=img_to_array(crop_face)
    crop_face=np.expand_dims(crop_face,axis=0)
    pred=maskmodel.predict(crop_face)[0][0]
    if pred==1:
        cv2.rectangle(img,(x,y),(x+l,y+w),(0,0,255),4)
    else:
        cv2.rectangle(img,(x,y),(x+l,y+w),(0,255,0),4)
st.image(img,channels='BGR',width=400)
elif(choice=='VIDEO'):
    file=st.file_uploader("Upload Video")
    maskmodel=load_model('mask.h5')
    window=st.empty()
    if file:
        tfile=tempfile.NamedTemporaryFile()
        tfile.write(file.read())
        vid=cv2.VideoCapture(tfile.name)
        i=1
        while(vid.isOpened()):
            flag,frame=vid.read()
            if flag:
                face=facemodel.detectMultiScale(frame)
                for (x,y,l,w) in face:

```



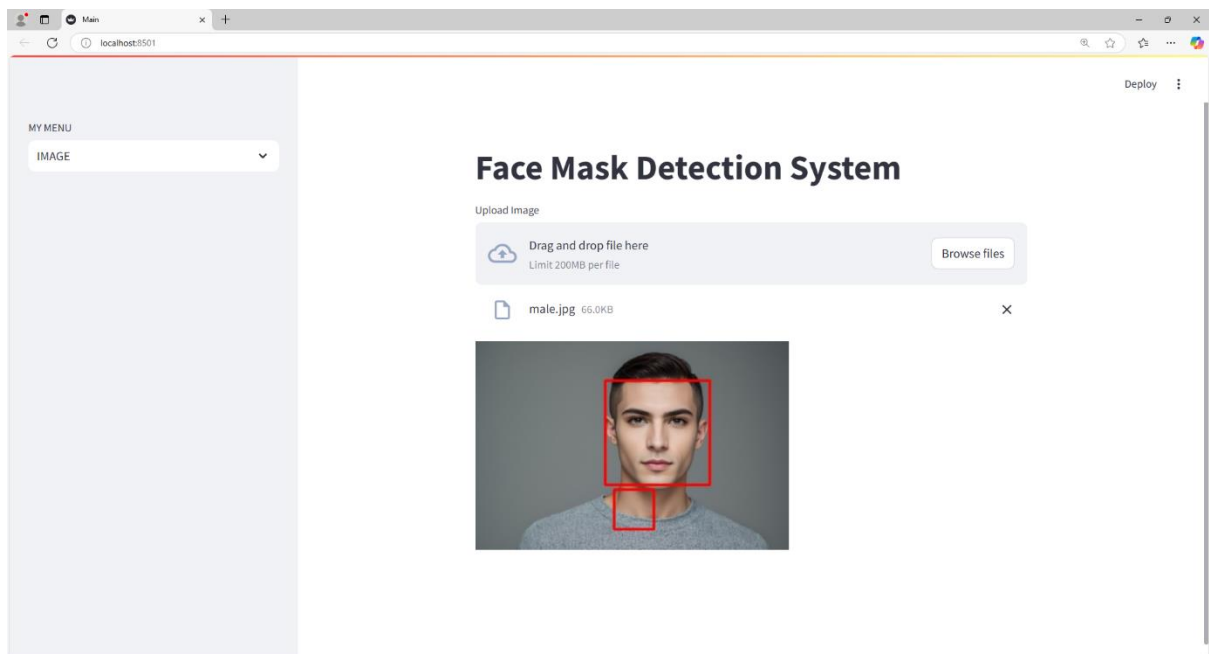
```

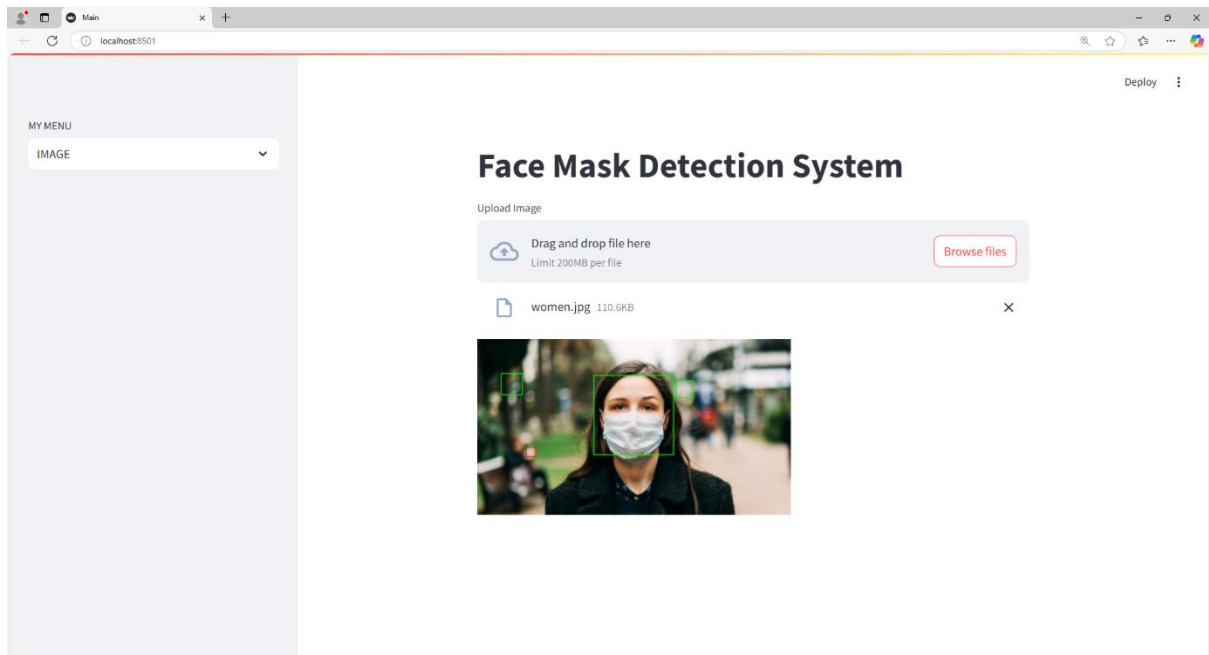
crop_face=load_img('temp.jpg',target_size=(150,150,3))
crop_face=img_to_array(crop_face)
crop_face=np.expand_dims(crop_face,axis=0)
pred=maskmodel.predict(crop_face)[0][0]
if pred==1:
    cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,255),4)
    path="data/"+str(i)+".jpg"
    cv2.imwrite(path,crop_face1)
    i=i+1
else:
    cv2.rectangle(frame,(x,y),(x+l,y+w),(0,255,0),4)
window.image(frame,channels='BGR')
btn2=st.button('Stop Camera')

```

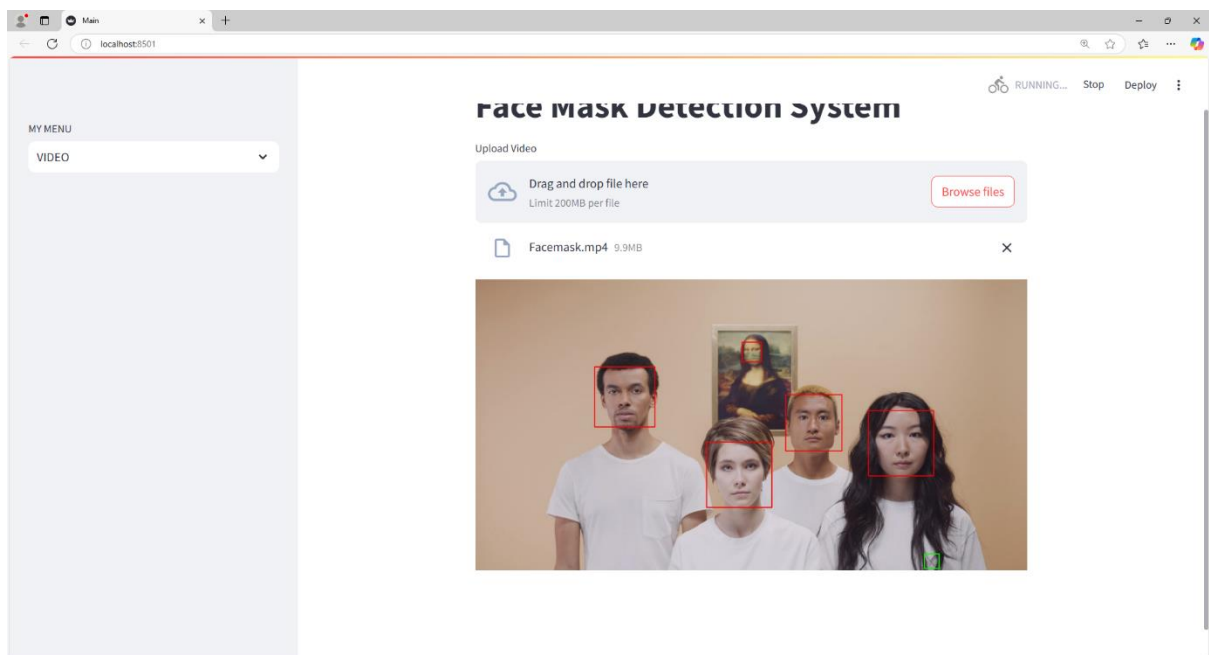
Output Screen :

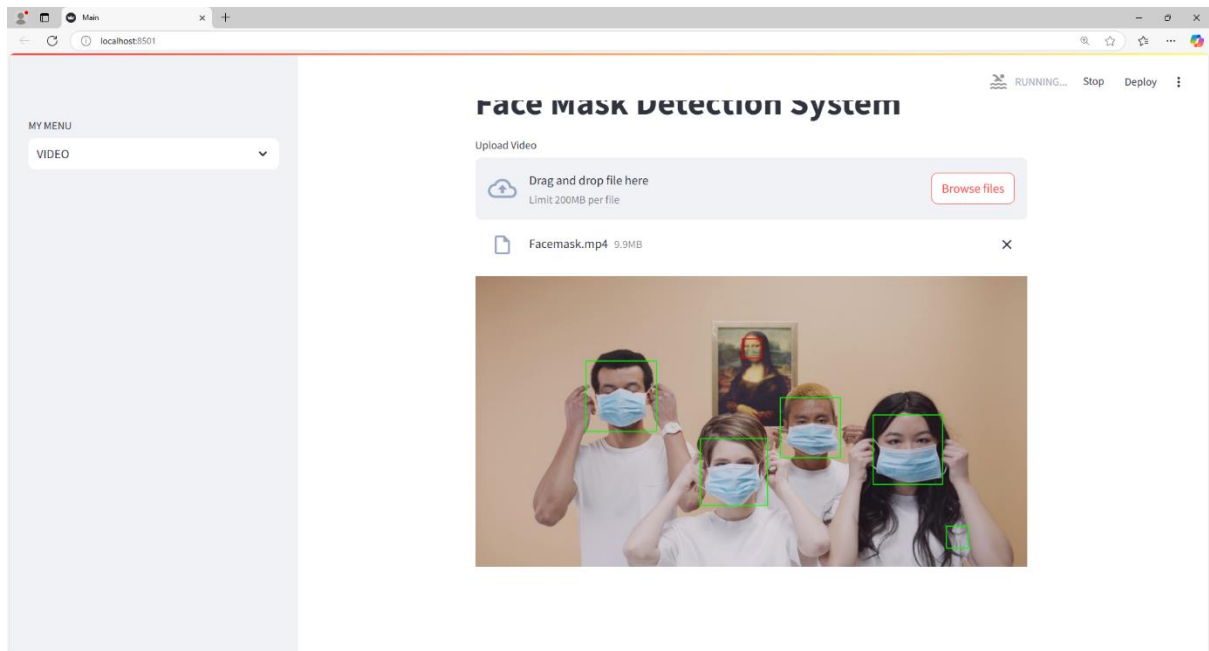
Detect From Image Source





Detect facemask by the source of Video





Detect By Camera Source

