**FACE MASK DETECTION SYSTEM**

WHAT

1.It is a computer vision application which can first detect the face into an image and then detect whether the face has a mask or not and then save the faces of no mask.

2.This application can get the image from different sources such as image, video, webcam, IP Camera

3.This application will be used using a browser on a Local Area Network.

WHY

1.This system can be used in many places such as hospitals, Research labs, Nuclear power Plant, Air borne Disease, High AQI Areas.

2.This kind of project shows your machine learning knowledge, programming skills, project development skills.

HOW

Backend: Code, Functionalities

Face Detection: OpenCV

Mask Detection: Tensorflow

Frontend: Interface

Web Application: Streamlit

1.Overview and OpenCV

2.Face Detection

3.Mask Detection

4.Frontend Development

5.Feature Integration

1.Create the virtual environment.

2.Activate the environment every time you use it.

Open CV

import cv2

#To read an image and show it.

img=cv2.imread("face.jpg")

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",img)

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

#To read an image and show it.

vid=cv2.VideoCapture("mask.mp4")

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

cv2.waitKey(41)

else:

break

cv2.destroyAllWindows()

import cv2

#To read an image and show it.

vid=cv2.VideoCapture("mask.mp4")

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

k=cv2.waitKey(41)

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

break

else:

break

cv2.destroyAllWindows()

import cv2

#To read an image and show it.

vid=cv2.VideoCapture(0)

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

k=cv2.waitKey(41)

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

break

else:

break

cv2.destroyAllWindows()

import cv2

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

vid=cv2.VideoCapture("mask.mp4")

pic=1

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.rectangle(frame,(x,y),(x+l,y+w),(255,255,255),8)

path="C:/project/faemask/Scripts/data/"+str(pic)+".jpg"

pic=pic+1

cv2.imwrite(path,frame[x:x+l,y:y+w])

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

cv2.waitKey(41)

else:

break

cv2.destroyAllWindows()

import cv2

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

vid=cv2.VideoCapture("face\_mask.mp4")

pic=1

while(vid.isOpened()):

count=0

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,0),8)

count=count+1

print(count)

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

cv2.waitKey(41)

else:

break

cv2.destroyAllWindows()

import cv2

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

vid=cv2.VideoCapture("http://192.168.0.100:8080")

pic=1

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.rectangle(frame,(x,y),(x+l,y+w),(255,255,255),8)

path="C:/project/faemask/Scripts/data/"+str(pic)+".jpg"

pic=pic+1

cv2.imwrite(path,frame[x:x+l,y:y+w])

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

cv2.waitKey(41)

else:

break

cv2.destroyAllWindows()

import cv2

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

vid=cv2.VideoCapture(0)

pic=1

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.rectangle(frame,(x,y),(x+l,y+w),(255,255,255),8)

path="C:/project/faemask/Scripts/data/"+str(pic)+".jpg"

pic=pic+1

cv2.imwrite(path,frame[x:x+l,y:y+w])

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",frame)

cv2.waitKey(41)

else:

break

cv2.destroyAllWindows()

**Tensorflow/Keras**

**Convolutional Neural Network**

Convolution Layer-> Max Pooling Layer ->Flatten Layer-> Dense Layer

#relu-rectified linear unit, if any input is negative relu converts it to zero and doesn’t allow the #neurons to get activated.

#sigmoid is used to predict the probability of a binary variable.

#because of the memory constraints, we rescale the images, the fact that bigger image usually #doesn’t provide a performance boost for CNN and lower batch size(this frequently causes longer #training time)

#flow\_from\_directory()- allows you to read the images directly from the directory and augment #them while the neural network model is learning on the training data.

#epochs-how many times you will have to train the model

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

from keras.models import Sequential

from keras.preprocessing.image import ImageDataGenerator #To organize the data properly

#Define the model

mymodel=Sequential()

#To add Layer

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(1,activation='sigmoid'))

mymodel.compile(optimizer='Adam',loss='binary\_crossentropy',metrics=['accuracy'])

#Organize the data

train=ImageDataGenerator(rescale=1./255)

test=ImageDataGenerator(rescale=1./255)

train\_set=train.flow\_from\_directory('train',target\_size=(150,150),batch\_size=16,class\_mode='binary')

test\_set=test.flow\_from\_directory('test',target\_size=(150,150),batch\_size=16,class\_mode='binary')

#Train the model

k=mymodel.fit(train\_set,epochs=10,validation\_data=test\_set)

#Save the model

mymodel.save('mask.h5',k)

#To read a face

import cv2

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

img=cv2.imread("man.jpg")

face=facemodel.detectMultiScale(img)

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

cv2.rectangle(img,(x,y),(x+l,y+w),(255,255,255),8)

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",img)

cv2.waitKey(0)

cv2.destroyAllWindows()

#To read whether a face has a mask or not for images:

import cv2

from keras.models import load\_model

from keras.utils import img\_to\_array,load\_img

import numpy as np

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

maskmodel=load\_model("mask.h5")

img=cv2.imread("mask2.png")

face=facemodel.detectMultiScale(img)

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

cv2.imwrite("temp.jpg",img[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(img,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(img,(x,y),(x+l,y+w),(0,255,0),8)

cv2.namedWindow("face window",cv2.WINDOW\_NORMAL)

cv2.imshow("face window",img)

cv2.waitKey(0)

cv2.destroyAllWindows()

main.py

import streamlit as st

st.set\_page\_config(page\_title="Mask Detection",page\_icon="https://m.media-amazon.com/images/I/61+ilDgVVwS.\_UL1500\_.jpg")

st.title("FACCE MASK DETECTION SYSTEM")

st.sidebar.image("https://cdn.hackernoon.com/images/oO6rUouOWRYlzw88QM9pb0KyMIJ3-bxfy3m27.png")

choice=st.sidebar.selectbox("MENU",("HOME","Image","IP Camera","WEB CAM"))

if(choice=="HOME"):

st.image("https://robogenesis.in/uploads/60d09d058c875a5d970563acbd3a089489ca0375b3b2e-min.gif")

st.write("Face Mask Detection System is a Computer Vision Machine Learning Application which can be accessed through IP Camera

and an detect whether the person is wearing a mask or not.")

elif(choice=="Image"):

st.markdown('<center><h2>IMAGE DETECTION</h2></center>',unsafe\_allow\_html=True)

main.py

import streamlit as st

import numpy as np

from keras.models import load\_model

from keras.utils import img\_to\_array,load\_img

import cv2

st.set\_page\_config(page\_title="Mask Detection",page\_icon="https://m.media-amazon.com/images/I/61+ilDgVVwS.\_UL1500\_.jpg")

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

maskmodel=load\_model("mask.h5")

st.title("FACCE MASK DETECTION SYSTEM")

st.sidebar.image("https://cdn.hackernoon.com/images/oO6rUouOWRYlzw88QM9pb0KyMIJ3-bxfy3m27.png")

choice=st.sidebar.selectbox("MENU",("HOME","Image","IP Camera","WEB CAM"))

if(choice=="HOME"):

st.image("https://robogenesis.in/uploads/60d09d058c875a5d970563acbd3a089489ca0375b3b2e-min.gif")

st.write("Face Mask Detection System is a Computer Vision Machine Learning Application which can be accessed through IP Camera

and an detect whether the person is wearing a mask or not.")

elif(choice=="Image"):

st.markdown('<center><h2>IMAGE DETECTION</h2></center>',unsafe\_allow\_html=True)

file=st.file\_uploader("Upload an image")

if file:

b=file.getvalue()

a=np.frombuffer(b,np.uint8)

img=cv2.imdecode(a,cv2.IMREAD\_COLOR)

face=facemodel.detectMultiScale(img)

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

cv2.imwrite("temp.jpg",img[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(img,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(img,(x,y),(x+l,y+w),(0,255,0),8)

st.image(img,channels='BGR')

elif(choice=='WEB CAM'):

k=st.text\_input("Enter 0 for Primary Camera or 1 for Secondary Camera")

btn=st.button("Start Camera")

if btn:

window=st.empty()

k=int(k)

vid=cv2.VideoCapture(k)

btn2=st.button("Stop Camera")

if btn2:

vid.release()

st.experimental\_rerun()

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.imwrite("temp.jpg",frame[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,255,0),8)

window.image(frame,channels='BGR')

elif(choice=='IP Camera'):

k=st.text\_input("Enter URL for the video")

btn=st.button("Start Camera")

if btn:

window=st.empty()

vid=cv2.VideoCapture(k)

btn2=st.button("Stop Camera")

if (btn2):

vid.release()

st.experimental\_rerun()

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.imwrite("temp.jpg",frame[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,255,0),8)

window.image(frame,channels='BGR')

main.py

import streamlit as st

import numpy as np

from keras.models import load\_model

from keras.utils import img\_to\_array,load\_img

import cv2

st.set\_page\_config(page\_title="Mask Detection",page\_icon="https://m.media-amazon.com/images/I/61+ilDgVVwS.\_UL1500\_.jpg")

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

maskmodel=load\_model("mask.h5")

st.title("FACCE MASK DETECTION SYSTEM")

st.sidebar.image("https://cdn.hackernoon.com/images/oO6rUouOWRYlzw88QM9pb0KyMIJ3-bxfy3m27.png")

choice=st.sidebar.selectbox("MENU",("HOME","Image","IP Camera","WEB CAM"))

if(choice=="HOME"):

st.image("https://robogenesis.in/uploads/60d09d058c875a5d970563acbd3a089489ca0375b3b2e-min.gif")

st.write("Face Mask Detection System is a Computer Vision Machine Learning Application which can be accessed through IP Camera

and an detect whether the person is wearing a mask or not.")

elif(choice=="Image"):

st.markdown('<center><h2>IMAGE DETECTION</h2></center>',unsafe\_allow\_html=True)

file=st.file\_uploader("Upload an image")

if file:

b=file.getvalue()

a=np.frombuffer(b,np.uint8)

img=cv2.imdecode(a,cv2.IMREAD\_COLOR)

face=facemodel.detectMultiScale(img)

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

cv2.imwrite("temp.jpg",img[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(img,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(img,(x,y),(x+l,y+w),(0,255,0),8)

st.image(img,channels='BGR')

elif(choice=='WEB CAM'):

k=st.text\_input("Enter 0 for Primary Camera or 1 for Secondary Camera")

btn=st.button("Start Camera")

if btn:

window=st.empty()

k=int(k)

vid=cv2.VideoCapture(k)

btn2=st.button("Stop Camera")

if btn2:

vid.release()

st.experimental\_rerun()

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.imwrite("temp.jpg",frame[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,255,0),8)

window.image(frame,channels='BGR')

elif(choice=='IP Camera'):

try:

k=st.text\_input("Enter URL for the video")

btn=st.button("Start Camera")

if btn:

window=st.empty()

vid=cv2.VideoCapture(k)

btn2=st.button("Stop Camera")

if (btn2):

vid.release()

st.experimental\_rerun()

while(vid.isOpened()):

flag,frame=vid.read()

if(flag):

face=facemodel.detectMultiScale(frame)

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

cv2.imwrite("temp.jpg",frame[y:y+w,x:x+l])

face\_img=load\_img("temp.jpg",target\_size=(150,150,3))

face\_img=img\_to\_array(face\_img)

face\_img=np.expand\_dims(face\_img,axis=0)

pred=maskmodel.predict(face\_img)[0][0]

if(pred==1):

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,0,255),8)

else:

cv2.rectangle(frame,(x,y),(x+l,y+w),(0,255,0),8)

window.image(frame,channels='BGR')

except:

pass

Screeshots:





