import os

import cv2

import numpy as np

from keras.preprocessing import image

import warnings

warnings.filterwarnings("ignore")

# rom keras.preprocessing.image import load\_img, img\_to\_array

from keras.models import load\_model

import matplotlib.pyplot as plt

import numpy as np

import tensorflow as tf

import statistics as st

# load model

model = load\_model("final\_model.h5")

face\_haar\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

i=0

GR\_dict={0:(0,255,0),1:(0,0,255)}

model = tf.keras.models.load\_model('final\_model.h5')

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

output=[]

cap = cv2.VideoCapture(0)

while (i<=50):

ret, img = cap.read()

#gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(img,1.05,5)

for x,y,w,h in faces:

face\_img = img[y:y+h,x:x+w]

resized = cv2.resize(face\_img,(224,224))

reshaped=resized.reshape(1, 224,224,3)/255

predictions = model.predict(reshaped)

# find max indexed array

max\_index = np.argmax(predictions[0])

emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'neutral', 'surprise')

predicted\_emotion = emotions[max\_index]

output.append(predicted\_emotion)

cv2.rectangle(img,(x,y),(x+w,y+h),GR\_dict[1],2)

cv2.rectangle(img,(x,y-40),(x+w,y),GR\_dict[1],-1)

cv2.putText(img, predicted\_emotion, (x, y-10),cv2.FONT\_HERSHEY\_SIMPLEX,0.8,(255,255,255),2)

i = i+1

cv2.imshow('LIVE', img)

key = cv2.waitKey(1)

if key == 27:

cap.release()

cv2.destroyAllWindows()

break

print(output)

cap.release()

cv2.destroyAllWindows()

final\_output = st.mode(output)

final\_output