Face detectection code

import cv2

import pickle

import numpy as np

import os

video=cv2.VideoCapture(0)

facedetect=cv2.CascadeClassifier('data/haarcascade\_frontalface\_default.xml')

faces\_data=[]

i=0

name=input("Enter Your Name: ")

while True:

ret,frame=video.read()

gray=cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces=facedetect.detectMultiScale(gray, 1.3 ,5)

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

crop\_img=frame[y:y+h, x:x+w, :]

resized\_img=cv2.resize(crop\_img, (50,50))

if len(faces\_data)<=100 and i%10==0:

faces\_data.append(resized\_img)

i=i+1

cv2.putText(frame, str(len(faces\_data)), (50,50), cv2.FONT\_HERSHEY\_COMPLEX, 1, (50,50,255), 1)

cv2.rectangle(frame, (x,y), (x+w, y+h), (50,50,255), 1)

cv2.imshow("Frame",frame)

k=cv2.waitKey(1)

if k==ord('q') or len(faces\_data)==100:

break

video.release()

cv2.destroyAllWindows()

faces\_data=np.asarray(faces\_data)

faces\_data=faces\_data.reshape(100, -1)

if 'names.pkl' not in os.listdir('data/'):

names=[name]\*100

with open('data/names.pkl', 'wb') as f:

pickle.dump(names, f)

else:

with open('data/names.pkl', 'rb') as f:

names=pickle.load(f)

names=names+[name]\*100

with open('data/names.pkl', 'wb') as f:

pickle.dump(names, f)

if 'faces\_data.pkl' not in os.listdir('data/'):

with open('data/faces\_data.pkl', 'wb') as f:

pickle.dump(faces\_data, f)

else:

with open('data/faces\_data.pkl', 'rb') as f:

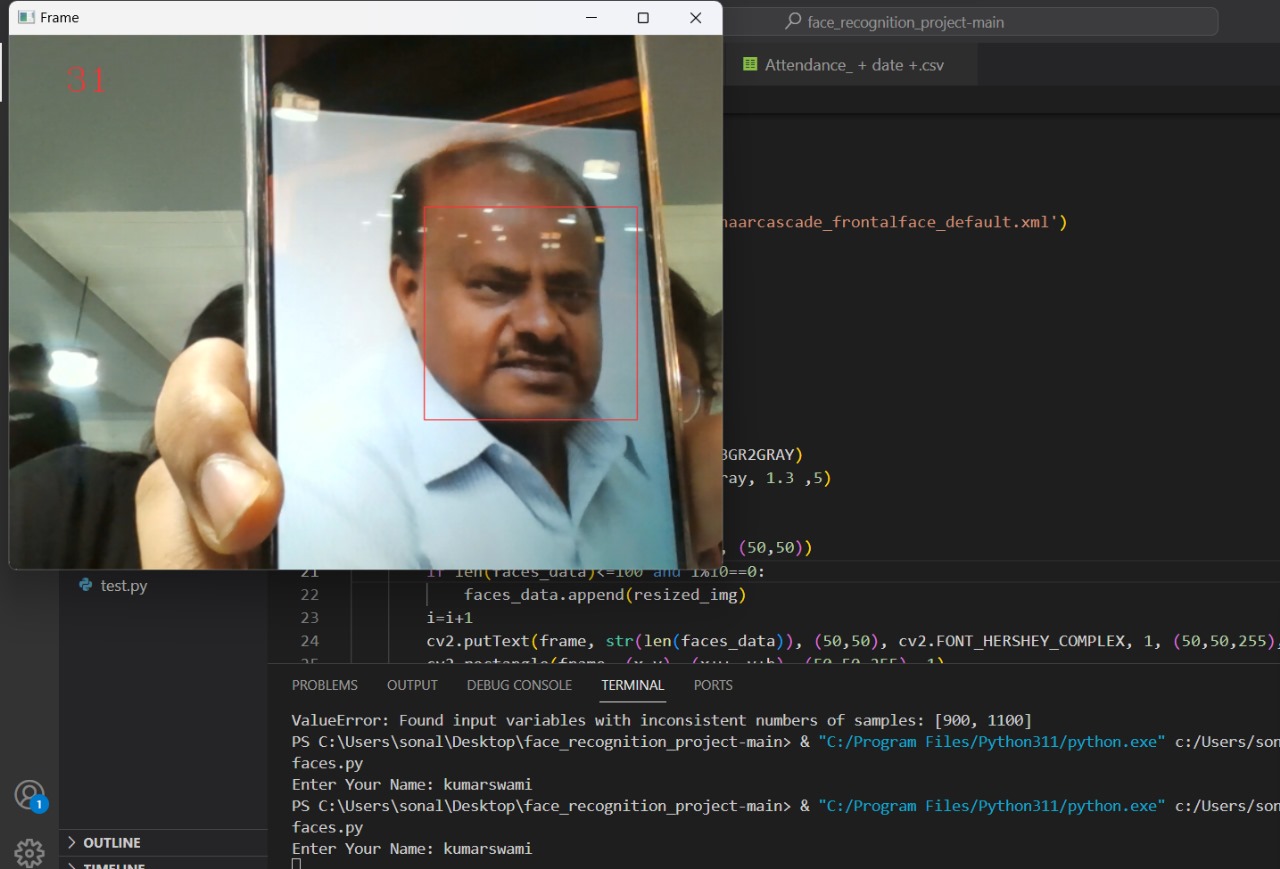
faces=pickle.load(f)

faces=np.append(faces, faces\_data, axis=0)

with open('data/faces\_data.pkl', 'wb') as f:

pickle.dump(faces, f)

Face detection result



Face recognition code

from sklearn.neighbors import KNeighborsClassifier

import cv2

import pickle

import numpy as np

import os

import csv

import time

from datetime import datetime

from win32com.client import Dispatch

def speak(str1):

speak=Dispatch(("SAPI.SpVoice"))

speak.Speak(str1)

video=cv2.VideoCapture(0)

facedetect=cv2.CascadeClassifier('data/haarcascade\_frontalface\_default.xml')

with open('data/names.pkl', 'rb') as w:

LABELS=pickle.load(w)

with open('data/faces\_data.pkl', 'rb') as f:

FACES=pickle.load(f)

print('Shape of Faces matrix --> ', FACES.shape)

knn=KNeighborsClassifier(n\_neighbors=5)

knn.fit(FACES, LABELS)

imgBackground=cv2.imread("background.png")

COL\_NAMES = ['NAME', 'TIME']

while True:

ret,frame=video.read()

gray=cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces=facedetect.detectMultiScale(gray, 1.3 ,5)

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

crop\_img=frame[y:y+h, x:x+w, :]

resized\_img=cv2.resize(crop\_img, (50,50)).flatten().reshape(1,-1)

output=knn.predict(resized\_img)

ts=time.time()

date=datetime.fromtimestamp(ts).strftime("%d-%m-%Y")

timestamp=datetime.fromtimestamp(ts).strftime("%H:%M-%S")

exist=os.path.isfile("Attendance/Attendance\_" + date + ".csv")

cv2.rectangle(frame, (x,y), (x+w, y+h), (0,0,255), 1)

cv2.rectangle(frame,(x,y),(x+w,y+h),(50,50,255),2)

cv2.rectangle(frame,(x,y-40),(x+w,y),(50,50,255),-1)

cv2.putText(frame, str(output[0]), (x,y-15), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255,255,255), 1)

cv2.rectangle(frame, (x,y), (x+w, y+h), (50,50,255), 1)

attendance=[str(output[0]), str(timestamp)]

imgBackground[162:162 + 480, 55:55 + 640] = frame

cv2.imshow("Frame",imgBackground)

k=cv2.waitKey(1)

if k==ord('o'):

speak("Attendance Taken..")

time.sleep(5)

if exist:

with open("Attendance/Attendance\_" + date + ".csv", "+a") as csvfile:

writer=csv.writer(csvfile)

writer.writerow(attendance)

csvfile.close()

else:

with open("Attendance/Attendance\_" + date + ".csv", "+a") as csvfile:

writer=csv.writer(csvfile)

writer.writerow(COL\_NAMES)

writer.writerow(attendance)

csvfile.close()

if k==ord('q'):

break

video.release()

cv2.destroyAllWindows()

attendance code

import streamlit as st

import pandas as pd

import time

from datetime import datetime

ts=time.time()

date=datetime.fromtimestamp(ts).strftime("%D-%m-%Y")

timestamp=datetime.fromtimestamp(ts).strftime("%H:%M-%S")

from streamlit\_autorefresh import st\_autorefresh

count = st\_autorefresh(interval=2000, limit=100, key="fizzbuzzcounter")

if count == 0:

st.write("Count is zero")

elif count % 3 == 0 and count % 5 == 0:

st.write("FizzBuzz")

elif count % 3 == 0:

st.write("Fizz")

elif count % 5 == 0:

st.write("Buzz")

else:

st.write(f"Count: {count}")

df=pd.read\_csv("Attendance/Attendance\_" + date + ".csv")

st.dataframe(df.style.highlight\_max(axis=0))