Faces\_add.py(adding face ficture to software along with name but in different pkl files)

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)

Test.py(recognition of registered face pictures)

from sklearn.neighbors import KNeighborsClassifier

import cv2

import pickle

import numpy as np

import os

video=cv2.VideoCapture(0)

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

with open('C:/Users/sonal/Desktop/face\_recognition\_project-main/data/names.pkl', 'rb') as f:

LABELS=pickle.load(f)

with open('C:/Users/sonal/Desktop/face\_recognition\_project-main/data/faces\_data.pkl', 'rb') as f:

FACES=pickle.load(f)

knn=KNeighborsClassifier(n\_neighbors=5)

knn.fit(FACES,LABELS)

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)

flattened\_img = np.array(resized\_img).flatten()

output = knn.predict([flattened\_img])

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)

cv2.imshow("Frame",frame)

k=cv2.waitKey(1)

if k==ord('q'):

break

video.release()

cv2.destroyAllWindows()

names.pkl and faces\_data.pkl

