A Project Report on

FACE RECOGNITION

Project report submitted to

Department of Computer Science
School of Computer Sciences

By

NITEN CHANDRA SETHY
2021MCA12
MCA II Semester

Under the Guidance of

R. GURURAJ MUKARAMBI
Assistant Professor

Central University of Karnataka
Kadaganchi, Kalaburagi - 585367, INDIA
September 3, 2022

CENTRAL UNIVERSITY OF KARNATAKA

Department of Computer Science

CERTIFICATE

This is to certify that the report entitled **FACE RECOGNITION** submitted by **NITEN CHANDRA SETHY** bearing Registration Number: **2021MCA12**, studying in MCA II Semester, to the Central University of Karnataka, Kadaganchi, Kalaburagi, is a record of the original project work carried out by him/her under my guidance and supervision during the V semester of MCA course. The work embodied in this project report has not been submitted fully or in part any where else.

Place: Kadaganchi
Date: September 3, 2022

DR. GURURAJ MUKARAMBI
Assistant Professor

Head, Department of Computer Science
School of Computer Sciences
Central University of Karnataka, Kadaganchi, Kalaburagi

DECLARATION

I hereby declare that the work embodied in this report entitled **FACE RECOGNITION**, submitted to Central University of Karnataka, Kadaganchi, Kalaburagi is a record of my original project work carried out by me in the Department of Computer Science, Central University of Karnataka, Kadaganchi, Kalaburagi, under the guidance of **DR. GURURAJ MUKARAMBI** , and that the full of part of this report has not been submitted to this or any other University or Institute.

Place: Kadaganchi
Date: September 3, 2022

NITEN CHANDRA SETHY
MCA II Semester
Central University of Karnataka
Kadaganchi, Kalaburagi

Acknowledgements

My sincere thanks to my teacher Mr.Gururaj Mukarambi,who gave me a golden opportunity to do this wonderful project of Computer Science on Face Recognition ,who also helped me completing my project. I came to know about so many things I am really thankful to him.

Place: Kadaganchi
Date: September 3, 2022

NITEN CHANDRA SETHY
MCA II Semester
Central University of Karnataka
Kadaganchi, Kalaburagi

Contents

1 Abstract

One of the simplest methods to distinguish each other's individual identities is to look at one other's faces. Face recognition is a type of personal identification system that uses a person's personal traits to determine their identity. The human face recognition mechanism is divided into two phases: face detection, which occurs relatively quickly in humans unless in situations where the item is positioned at a short distance away, and introduction, which recognizes a face as an individual. The face recognition mini-project undertaken for the visual perception and it describes the technologies in the Open-Computer-Vision (OpenCV) library as well as the methods for implementing them in Python. For face detection, Haar-Cascades is used.

2 Introduction

Facial recognition is a method of recognizing or verifying a person's identification by looking at their face. People can be identified in pictures, films, or in real-time using facial recognition technology. Face recognition is the task of identifying an already detected object as a known or unknown face. Face recognition is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Facial recognition is a category of biometric security.

Tools/software used:

a) PyCharm: PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.
b) OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It was built

to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.
c)Camera: This will be used to the real time face recognition

3 Scope

Face recognition technology is being used all over the world, and it is having a positive impact. This technology has a lot of potential in India, and it can help the country progress in a variety of ways.
The technology and its applications can be used in a variety of industries across the country.
–Preventing the fraud at ATMs in India. A database of all customers with ATM cards in India can be created and facial recognition systems can be installed. So, whenever a user will enter in ATM his photograph will be taken to permit access after it is matched with the stored photo from the database.
–Reporting duplicate voters in India.
–Passport and visa verification can also be done using this technology.
–For access control verification and identification of authentic users it can also be installed in bank lockers and vaults.

4 Literature Survey

-Face Detection and Recognition using OpenCV and Python:
Tejashree et al [1] explained, according to, this research paper gives an ideal way of detecting and recognizing human faces using OpenCV, and python which is part of deep learning. This report contains ways in which deep learning an important part of the computer science field can be used to determine the face using several libraries in OpenCV along with python. This report will contain a proposed system that will help in detecting the human face

in real-time. This implementation can be used on various platforms in machines and smartphones, and several software applications. They have used neural networks, Hidden Markov model, face matching done geometrically, and template matching. They also mentioned that using HMM model gave an 87 percent accuracy as it always gives out the best and right choice of face detection through the stored dataset.

-Designing of Face Recognition System

Vijay Kumar et al.[2] described that, The main focus of the paper is to design a face recognition system for advanced applications such as access and security, payments, criminal identifications, etc. To improve the calculation speed and efficiency, Haar like features are used in face detection and face recognition is done by using LBPH because of its computational simplicity and accuracy .The method of identification will be based on face recognition which is further divided into three steps: detection of the face, extractions of the features and classification, and real-time recognition, which is predicated on the Viola-Jones object detection algorithm that uses AdaBoost classifier with Haar and LBP features. and implemented using Python using the OpenCV library. the primary process of the system is to store the number of images in its database. This database is used to identify the related person. If the camera captured any face, then it calls the database and checks the measurements available in its database, and they also mentioned that efficiency is about 80 percent. they have been tested in different environments and lighting conditions with different cameras and the results are near about same.

-Live video face recognition:

Deshak Bhatnagar[3] explained that this section outlines the work already done in video facial recognition and general facial recognition. The literature here focuses primarily on facial recognition and tasks that already exist in this area. Also, look at the angle of view of video face recognition, more specifically live video face recognition. Here he used LBPS(Local Binary

Pattern Histogram) to recognize the face. To do this, we first need to tell the machine what the face will look like so that you can recognize at least one face in the video. To train on a machine what a face looks like, we need to create a cascade in the XML language with the addresses of thousands of images of the different faces used to train the machine. With the help of these cascades, the machine will know the human face to recognize in this case. He also concluded that Rigorous experimental / implementation process results for live VideoFace recognition were over 95 percent.

-Face Recognition Using Opencv and Python:

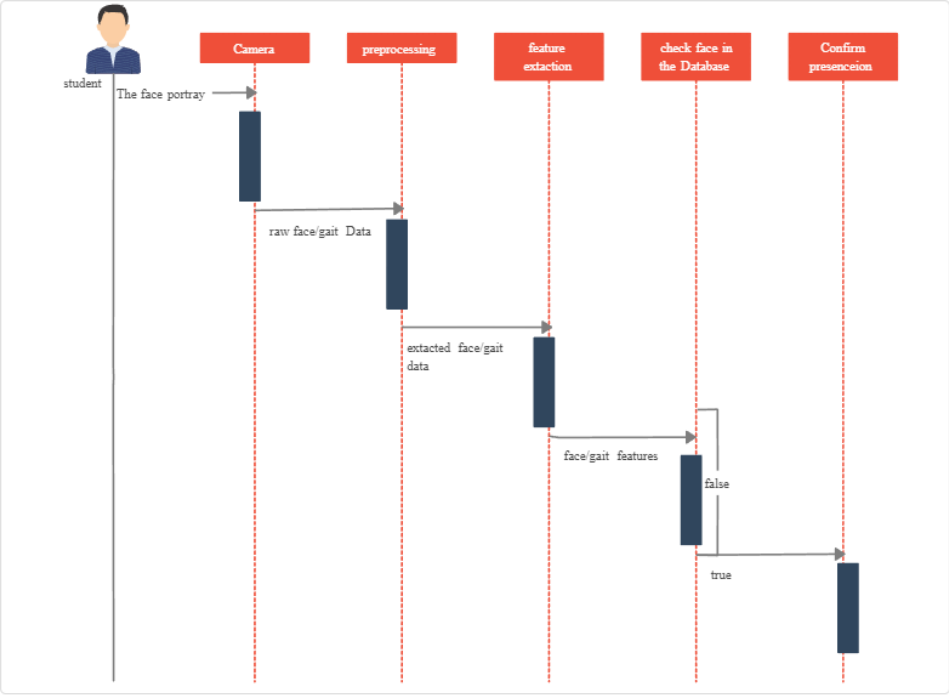
Rajesh Kumar Reddy.C et al.[[4] described that this paper presents an approach for extracting photographs. An attribute that details someone's face. The proposed method involves cascading object detection. this The method is efficient and effective considering each image. Face recognition is one of the leading biometric technologies. It's the most attractive and Successful application of pattern recognition and image analysis. Normally the input image of the method is rescaled Face recognition. Facial recognition systems have two main tasks: verification and identification. Verification means 1:1 A match that compares the face image to the face image in the template for which the ID is required. ID This means it's a 1: N issue comparing the query face image to all template images in the face database.

-Face recognition using PCA

Liton Chandra Paul et al.[5] described in his paper is mainly related to the construction of face recognition systems using Principal Component Analysis (PCA). PCA is a statistical approach used to reduce the number of face recognition variables. In PCA, each image in the training set is represented as a linear combination of weighted eigenvectors called eigensurfaces. These eigenvectors are obtained from the covariance matrix of the training image set. The weight is determined after selecting the most relevant set of eigenfaces. Recognition is done by projecting a test image into a subspace that spans the eigenplane, and then by measuring the minimum Euclidean dis-

tance. A series of experiments were conducted to evaluate the performance of facial recognition systems. The goal of this study is to research and develop an efficient MATLAB program for face recognition using principal component analysis, and to perform program optimization and accuracy testing. This approach is preferred because of its simplicity, speed, and learning ability.

4.1 Data Flow Diagram



5 Methodology

Here we have the methodology and descriptions of the applications used for data gathering, face detection, training and face recognition. The project was coded in Python using PYCharm IDEs. First stage was creating a face detection system using Haar-cascades. Although, training is required for creating new Haar-cascades, OpenCV has a robust set of Haar-cascades that was used for the project. Using face-cascades alone caused random objects to be identified and obtain stable face detection. A camera object is created using the cv2.VideoCapture() to capture images. By using the CascadeClassifier.detectMultiScale() object of various sizes are matched and location is returned.

Face Recognition Process

For this project I have used Linear binary pattern histograms algorithms. There are three stages for the face recognition as follows:

-detection and collection of data set

-training of the data

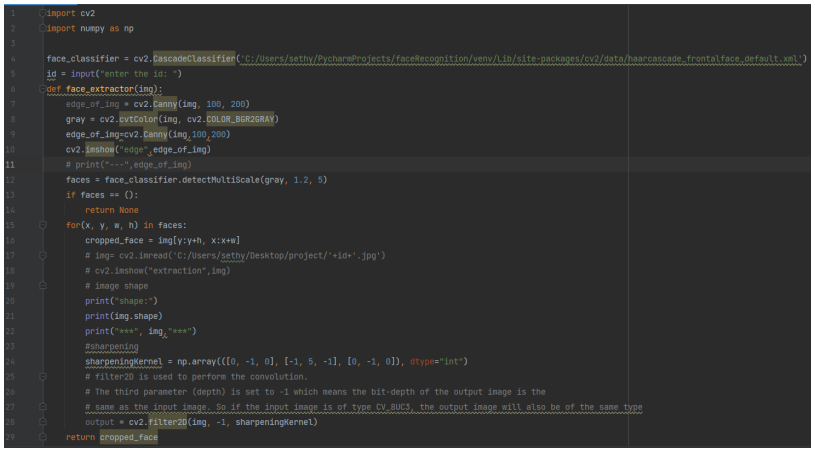
-recognition

Collecting the image data: In this section we have to collect various image data. Since this a very time taking task, I have used web cam for the data set collection purpose which will automate and will collect many images within few seconds.

The objective of this section is to detect object of face in real time and to keep tracking of the same object.

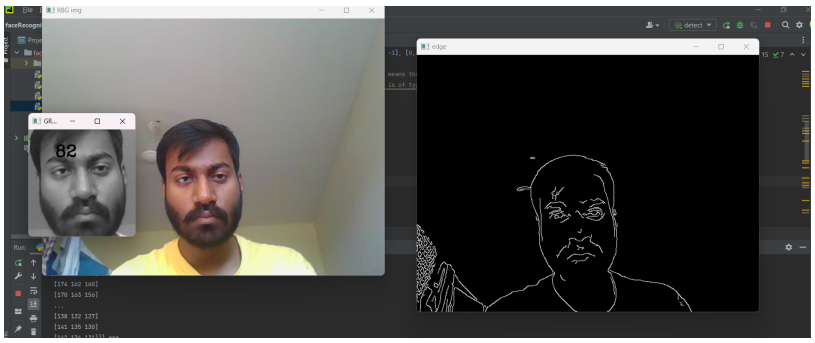
Code for data set collection: Here I have use cascade classifier frontalface-default.xml file, which is already present inside the python library. Then, I defined a method to do face feature extraction. In this face extraction I am converting image to gray scale and using detectMultiScale (a function for detection both faces and objects. In order for the function to search for exactly the faces, we pass it to the corresponding cascade.)

During taking input i have done some image processing works like gray scaling , re-sizing image, removeing blurness and sharpening of images. All these work happened in between taking images from web cam and storing in the desired location. And also I an extracting the facial coordinates which will be helpful for training section.
Sample code for the data set collection from web cam is given :

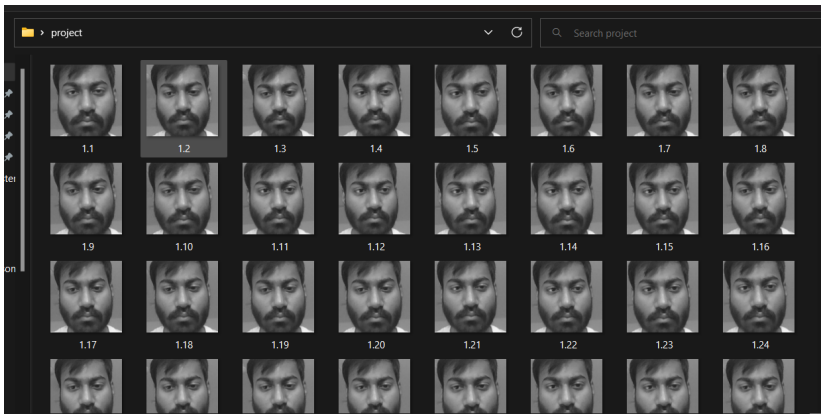


```python
import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier('C:/Users/sethy/PycharmProjects/faceRecognition/venv/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml')
id = input("enter the id: ")
def face_extractor(img):
    edge_of_img = cv2.Canny(img, 100, 200)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    edge_of_img=cv2.Canny(img,100,200)
    cv2.imshow("edge",edge_of_img)
    # print("---",edge_of_img)
    faces = face_classifier.detectMultiScale(gray, 1.2, 5)
    if faces == ():
        return None
    for(x, y, w, h) in faces:
        cropped_face = img[y:y+h, x:x+w]
        # img= cv2.imread('C:/Users/sethy/Desktop/project/'+id+'.jpg')
        # cv2.imshow("extraction",img)
        # image shape
        print("shape:")
        print(img.shape)
        print("+++", img,"+++")
        #sharpening
        sharpeningKernel = np.array(([0, -1, 0], [-1, 5, -1], [0, -1, 0]), dtype="int")
        # filter2D is used to perform the convolution.
        # The third parameter (depth) is set to -1 which means the bit-depth of the output image is the
        # same as the input image. So if the input image is of type CV_8UC3, the output image will also be of the same type
        output = cv2.filter2D(img, -1, sharpeningKernel)
    return cropped_face
```

```python
cap = cv2.VideoCapture(0)
count = 0

while True:
    ret, frame = cap.read()
    if face_extractor(frame) is not None:
        count = count + 1
        cv2.imshow('RBG img', frame)
        face = cv2.resize(face_extractor(frame), (200, 200))

        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)

        file_name_path = 'C:/Users/sethy/Desktop/project/'+str(id)+"."+str(count)+'.jpg'

        cv2.imwrite(file_name_path, face)

        cv2.putText(face, str(count), (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow('GRAY img', face)
        print("face coorinates",str(count), face)
    else:
        print("Face not found")
        pass

    if cv2.waitKey(1) == 13 or count == 100:
        break

cap.release()
cv2.destroyAllWindows()
```



Data stored in file as png format:



Training of the data:

In this section train of collected data is done using LBPH algorithm.

LBPH (Local Binary Pattern Histogram) is a Face-Recognition algorithm it is used to recognize the face of a person. It is also known for its performance and how it is able to recognize the face of a person from the both front face and side face.

All images are represented in the Matrix formats,which are composed of rows and columns. The basic components of an image is the pixel and an image is made up of a set of pixels.The value of pixels ranges between 0 to 255 for every image.And through the train method we can remove the images that are not suitable for training.

Sample code is given below:

```python
import cv2
import numpy as np
from PIL import Image
import os

data_path = 'C:/Users/sethy/Desktop/project/'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier('C:/Users/sethy/PycharmProjects/faceRecognition/venv/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml');
# function to get the images and label data
def getImagesAndLabels(path):
    imagePaths = [os.path.join(data_path,f) for f in os.listdir(data_path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # grayscale
        img_numpy = np.array(PIL_img,'uint8')
        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)
        for (x,y,w,h) in faces:

            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
    return faceSamples,ids
print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
```

```python
print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces, ids = getImagesAndLabels(data_path)
recognizer.train(faces, np.array(ids))
# Save the model into trainer/trainer.yml

#recognizer.write('C:/Users/sethy/Desktop/trainer/.yml')
# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))

print("Dataset Model Training Completed ")
```

Recognition:

This section is used for recognizing face from the data set. In first I set the path where the data set is stored then re-trained it and created a face detector method in which I am detecting face ,gray scaling etc are done . The recognition is done by detecting face then comparing the coordinates from the live face with the stored data set in the file .if the coordinates match then it will show the name if it doesn't match then it will show unknown face , if face is not detected then it will show face not found. I am using predict method to find the confidence of the matching of coordinates.

Below sample code is given

```python
import cv2
import numpy as np
from os import listdir
from os.path import isfile, join
import matplotlib .pyplot as plt


data_path = 'C:/Users/sethy/Desktop/project/'
onlyfiles = [f for f in listdir(data_path) if isfile(join(data_path,f))]
acc=[]
Training_Data, Labels = [], []

for i, files in enumerate(onlyfiles):
    image_path = data_path + onlyfiles[i]
    images = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    Training_Data.append(np.asarray(images, dtype=np.uint8))
    Labels.append(i)

Labels = np.asarray(Labels, dtype=np.int32)

model = cv2.face.LBPHFaceRecognizer_create()

model.train(np.asarray(Training_Data), np.asarray(Labels))

print("Dataset Model Training Complete!!!!!")

face_classifier = cv2.CascadeClassifier('C:/Users/sethy/PycharmProjects/faceRecognition/venv/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml')
```

```python
def face_detector(img, size = 0.5):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,5)

    if faces == ():
        return img,[]

    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y),(x+w,y+h),(0,255,0),2)
        roi = img[y:y+h, x:x+w]
        roi = cv2.resize(roi, (200,200))

    return img,roi

cap = cv2.VideoCapture(0)
while True:

    ret, frame = cap.read()

    image, face = face_detector(frame)

    try:
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
        result = model.predict(face)
```

.

.

.

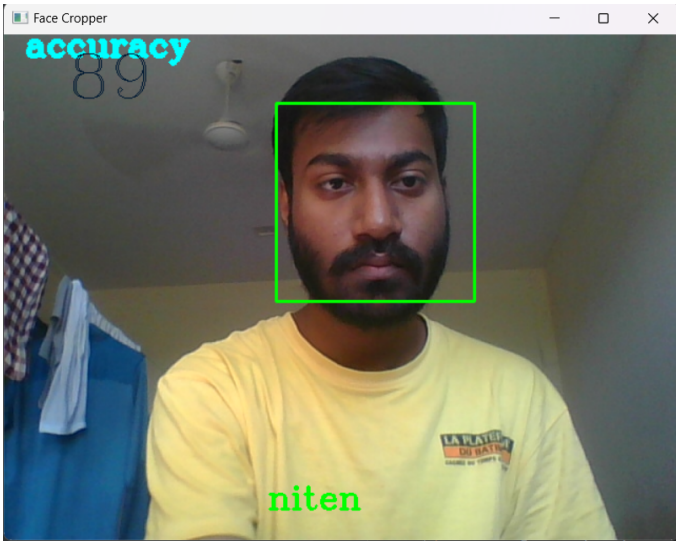
.

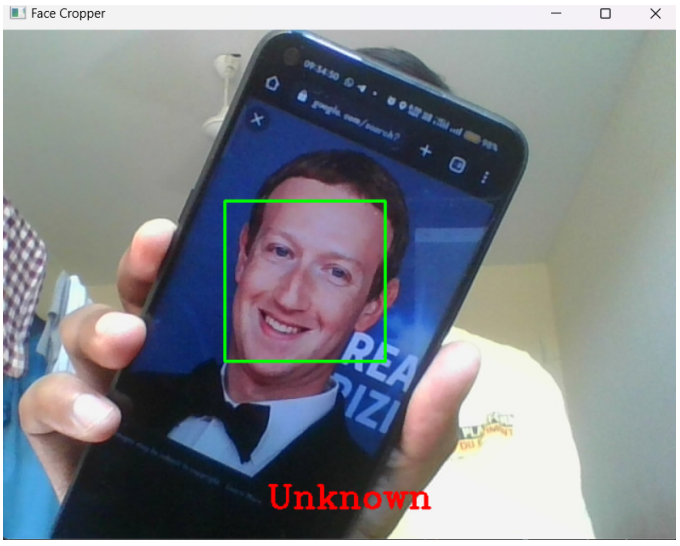
.

```
        if result[1] < 500:
            confidence = int(100*(1-(result[1])/300))



        if confidence > 82:
                cv2.putText(image, "niten", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2
                cv2.putText(image,"accuracy",(20, 20), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 0),
                cv2.putText(image, str(confidence),(60,60), cv2.FONT_HERSHEY_COMPLEX, 2, (50, 25, 0
                cv2.imshow('Face Cropper', image)
                acc.append(confidence)
        else:
            cv2.putText(image, "Unknown", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
            cv2.imshow('Face Cropper', image)


    except:
        cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0),
        cv2.imshow('Face Cropper', image)
        pass

    if cv2.waitKey(1)==13:
        break
```

```
77
78      plt.plot(acc)
79      plt.ylabel('accuracy')
80      plt.xlabel('run time')
81      plt.title('confidence')
82      # plt.imshow('acc',acc)
83      plt.show()
84
85      cap.release()
86      cv2.destroyAllWindows()
```

Live input and output:


Face Cropper
accuracy
89
niten


Face Cropper
Unknown

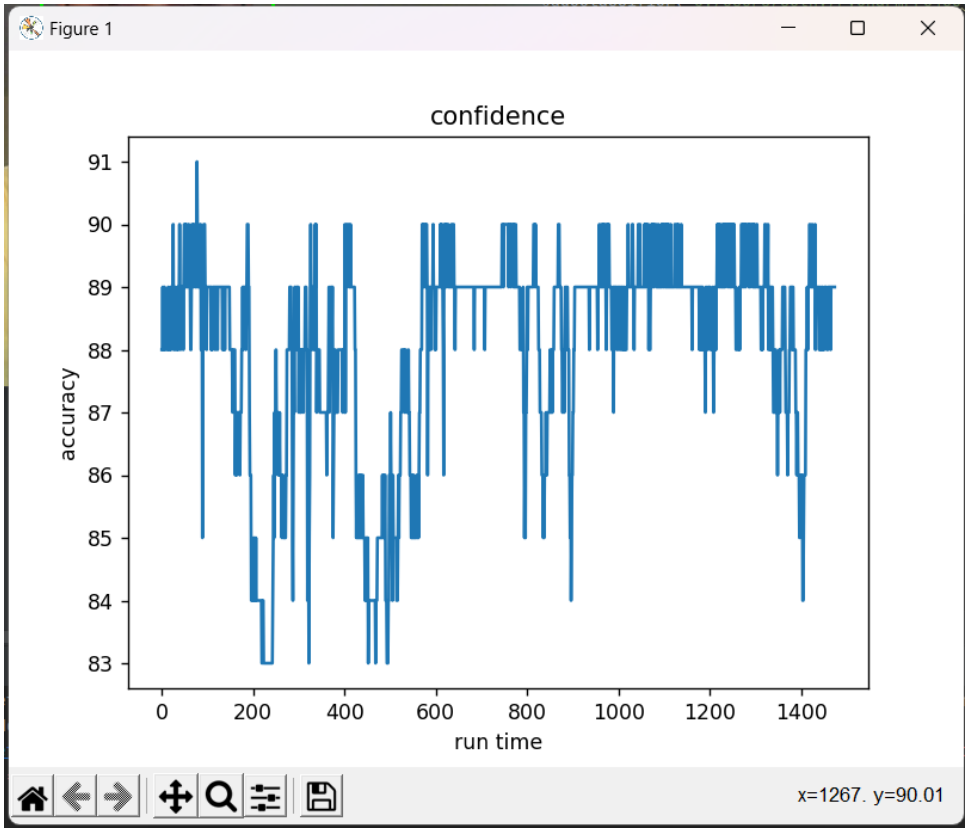


Figure 1
confidence
accuracy
run time
91
90
89
88
87
86
85
84
83
0
200
400
600
800
1000
1200
1400
x=1267. y=90.01

6 Uses/applications of face recognization:

This technology is used for a variety of purposes. These include:
-Unlocking phones
-Finding missing persons
-Tracking student or worker attendance
-Automobile Security

7 Advantages and Disadvantages

Advantages
–Helps find missing people.
–Better security.
–Automated identification.
Disadvantages
–Huge storage is needed
–Technology is still new
–Expensive

8 Future Scope

Facial recognition technology has a promising future. Forecasters predict that this technology will grow at a rapid pace and create significant money in the next years. The primary categories that will be heavily influenced are security and surveillance. Private companies, public structures, and schools are among the other domains that are now welcome it with open arms. It is expected that businesses and banking institutions would embrace it in the future years to prevent fraud in debit/credit card purchases and payments, particularly those made online. This method would close the gaps in the widely used yet insecure password system. Robots with facial recognition technologies may potentially make an appearance in the future. They can be beneficial in a variety of situations. They can assist in the completion of tasks that are inconvenient or difficult for humans to do.

9 Conclusion

I finally concluded that face recognition is a new technology that has a lot of potential. Face recognition may help businesses save money and time, as well as establish new revenue streams, if done correctly.

10 Plagarism Report

Report Title:	Face Recognition
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/90502bb94eba158302d433e36bd5429039c231662218143
Report Generated Date:	03 September, 2022
Total Words:	1803
Total Characters:	12485
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	91%
Matched:	9%

References

https://www.kaspersky.com/resource-center/definitions/what-is-facialrecognition https://www.techsciresearch.com/blog/future-of-facial-recognitiontechnology/105.html https://docs.opencv.org/4.0.0/d7/d8b/tutorialpyfacedetection.html https://www.pathpartnertech.com/challenges-faced-by-facialrecognition-system/