Face Recognition Systems for Modern Way of Security

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
AWARD DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEM

Submitted by:
MohammedAhmed Ibrahim
MohammedAhmed Idris
(2K20/ISY/25)

Under the Supervision of:
Dr. PRIYANKA MEEL
(Assistant Professor)

DEPARTMENT OF INFORMATION TECHNOLGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042
MAY, 2022

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Main Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, MohammedAhmed Ibrahim MohammedAhmed Idris, Roll No. 2K20/ISY/25 student of M.Tech in Information Systems, hereby declare that the Major Project II titled "Face Recognition System for Modern Way of Security" that which is submitted by my humble personality to the Department of Information Technology at Delhi Technological University - Delhi in partial fulfillment of the requirements for the award of the degree of Master of Technology, this project is original and without proper citation is not copied from any sources. This work has not previously formed the basis for the award of any Degree, Diploma Associate ship, Fellowship or other similar title or recognition.

Place: DELHI
Date: May, 2022

MohammedAhmed Ibrahim MohammedAhmed Idris
(2K20/ISY/25)

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Main Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Major Project II titled “Face Recognition System for Modern Way of Security” which is submitted by MohammedAhmed Ibrahim MohammedAhmed Idris, Roll No. 2K20/ISY/25 department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: May, 2022

Dr. Priyanka Meel
SUPERVISOR
Assistant Professor

Abstract

In this difficult time full of obstacles and with the rapid spread of the Internet we find that the Artificial Intelligent, Internet of Things technology is hot and at the same time the biometric detection technology represented by face recognition technology will attract more attention. Face recognition technology includes large number and different types of applications, including security, personal authentication; secure internet connection, and computer entertainment. Face recognition technology is based on knowledge of the human facial features to identify and help train machine learning systems to meet the needs of artificial intelligence. The current known biotechnologies technology, facial recognition, attention to fingerprints, palm printer attention, awareness of the iris or retina, and voice recognition through visual acuity, these technologies may work. Among them, facial recognition, as the most widely used and appropriate biotechnology has clear advantages in terms of improved safety and no errors that require minimal consideration compared to other biometric assurance techniques, ease of integration costs and existing safety features, ease of use and general acceptance. In addition, for most of the services currently offered, the user authentication process can be done using this technology due to the high level of accuracy it provides in finding and detecting faces. It has excellent solutions in the public and commercial sectors. In these current years it has seen significant progress in this area due to the development of facial modeling and deep analysis techniques. However, there are many safety issues associated with this technology, whether on the part of the service provider and its user that we will try to enlighten on in this research work.

It is expected that this new system is designed with strong safeguards to protect privacy. It is clear that institutions will obtain permission to enter this system through partnership agreements, and there will be clear restrictions on the ability of the private sector to access the data within the system.

Keywords: Face recognition, Face detection, Verification, Identification.

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the Almighty My God ‘Allah’, for His showers of blessings throughout my research work to complete this project successfully.

Countless people supported my effort on this project that With God's help would not have been possible without them; I would like to express my deepest thanks to my respected supervisor **Dr. Priyanka Meel** (Assistant Professor, Department of Information Technology) and all the faculty members of the Department of Information Technology at DTU. They all provided me with immense support and guidance for the project and also giving me the opportunity to do this research and providing invaluable guidance throughout this research.

Very special thanks to the M.Tech students of Information System, for their support during my research work. I also thank all the staff of Research section of Delhi Technological University for their kindness.

I would also like to appreciate the support provided to us by our seniors and our peer group who aided us with all the knowledge they had regarding various topics. Finally, my thanks go to all the people who have supported me to complete the Project work directly or indirectly.

MohammedAhmed Ibrahim
MohammedAhmed Idris
(2K20/ISY/25)

Contents

List of Figures

CHAPTER 1

1. INTRODUCTION AND MOTIVATION

Face recognition technology is based on knowledge of the human facial features to identify and help train machine learning systems to meet the needs of artificial intelligence. Right now in this present time it is considered that it’s very necessary to do preserve the protection process of people’s information or property is becoming more important because people are accustomed to store their precious information in their smart phones and if unfortunately their movable will stole or misplace then an outright breach of privacy resulting that all their important information are going to be access by someone else un authorized. As we hear about many of the crimes related to MasterCard fraud, computer damage by hackers, etc. All frauds through with the help of fundamental flaw and regular information of user for accessing in most of the cases like ID cards, weak passwords, PIN, keys, etc. None of above information is de facto defining us; they’re means to authenticate us. To forestall, many applications are required. We all want fast, easy and accurate identification and access. Rather than paper work in this era, technology has become essential to work electronically. This growth of digital transformation in E-Commerce has leaded resulted to increasing demand because it is fast tracking and accurate user identification and authentication.

Face Recognition could be a recognition technique as an advanced measuring method that has attracted growing interest within the optimal usage of digital images and videos in many applications accustomed detect faces of people whose images saved within the datasets, it working by analyzing external body part in picture then transforming it into digital data format in line with features for extracting in each face like gap between left and right eye, Nose specification, spacing of ears and therefore the chin width, then matching specific face with an image of device’s owner to determine if an individual, or in other sectors with pictures in a database of faces. And we don't forget either Face Recognition growing public worry for security purposes, it require for verify the identification in the digital world. Last decade has provided rapid and noticeable advancement in this field because of advances in facial modeling and analysis techniques advanced for facial detection and tracking, reliable facial recognition still offers massive challenges.

With the growing interest in technologies associated with social distancing, we find that the biometric surveillance technologies may prove useful in slowing the spread of COVID-19 pandemic and leads to facial recognition systems by adopting expanding in this case. Governments are encouraging biometrics companies by developing and implementing facial recognition technology can building confidence now through genuine communication of the possibilities of the technology in a way that allow to all members of society to access it.

Since the COVID-19 spread through direct communication, the previously used unlocking system supported fingerprint or traditional passwords is which requires direct touch devices. By latest there are multiple applications that depend on facial recognition, such as face attendance, face access control; face authentication based mobile payment, face gates at metro/ train stations, community-based security surveillance, public access and exit points, etc.

1.1 Identification and Verification

We can usually divide facial recognition technology in two matching methods of the captured biometric feature first part is Verification (Confirmation). And the second part is Identification (Recognition).

The process of starting a face authentication system involves individual identification to allow or reject a personal ID claim. This method comparing the captured face image with the personal imagery stored within the database. If the user's image that put in to the system is that recognized, claiming to be authorized means that the system will accept that customer, furthermore the system will refuse that customer (fraudster). Knowing that there are many applications that claim face verification mode, such as smart phones or computer login, E-gate control.

The second process is face Identification or recognition that may include one of many similarities; the system must contact the user's identity from those registered or declare anonymous.
In all cases, the system must have a referential location (Gallery) containing all the vector features (signatures) of the faces of people thought to be known by the system.

In short, face recognition technology performs three basic functions:

- Detection: Detect a face in a photo.
- Verification: Verification of identity associated with that face.
- Identification: Matching a photo of an unknown face with a gallery of celebrities.

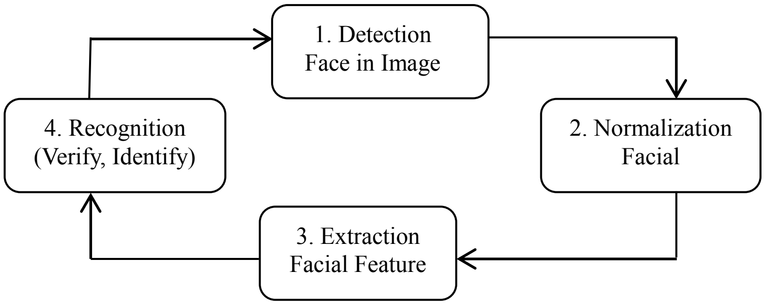


Figure 1.1: Steps in the Face Recognition Process

The final identification system showing can be rated using an Identification Rate (IdR). The output data from IdR is straight forward: Ratio of previously registered subjects successfully set to the proper identity.

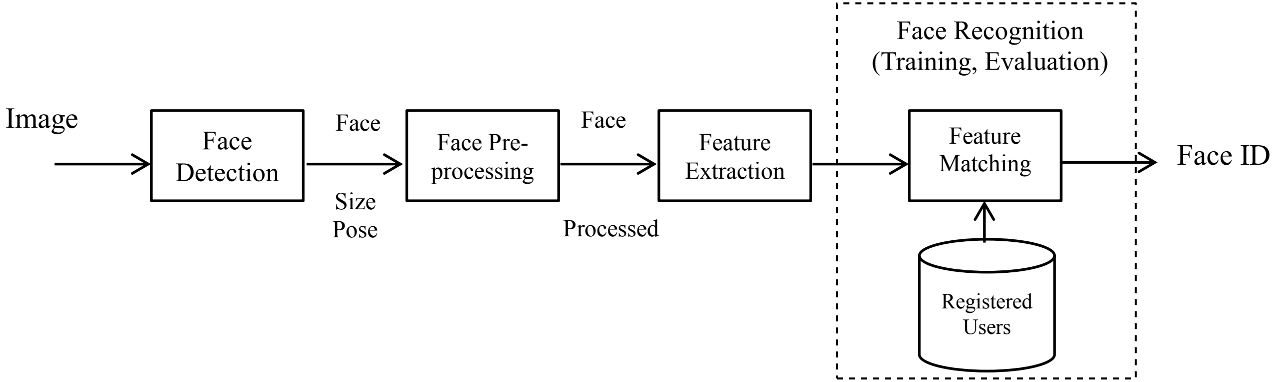


Figure 1.2: Structure of a Face Recognition System

1.2 Face Recognition Technology

Face recognition is considered as one of the foremost promising identity verification methods probably because it's the foremost naturalistic thanks to recognize the identity among personalities. Individual faces act one among the mostly mutual visual patterns in our surroundings. Thus, it is usual for people to spot individual by his face, however it can be not possible to founding it through means of his fingerprint, iris, retina or perhaps a private mark, because of the massive quantities of existing languages, for even several individuality in worldwide.

Facial recognition seems easy task, but it is not as it looks. There is a lot of programming and mathematics include behind it.

Facial recognition in its simplest form is a trend of exploring an individual's face through technology. Biometrics and security face recognition system uses to map facial features from various sources whether picture or video. It takes the information and checks the details of important information of registered users with faces stored in database to detect a perfect corresponding that can assist us to prove individual identity, however as well raising privacy concerns.

Face Recognition Algorithms

- Basically the face recognition algorithm sees data from which the features of a vector face are obtained through feature modeling, that data can be stored and accessed. And the recognition results are obtained by marking the classifier.
- The key here is how to get the distinctive features of different faces, usually when we get to know someone, we will look at the shape of the eyebrow, the contour of the face, the shape of the nose, the type of the eye, etc.
- A face recognition algorithm engine (Training) must be trained to get discriminative features.

1.3 Stages of the Developing Face Recognition Algorithms

The face recognition algorithm went through three stages from the earliest algorithms, Synthetic features / workbooks, and deep learning. Currently, deep learning algorithms are prevalent, which has greatly improved the accuracy of face recognition, prompting this technology to be practical.

1.3.1 The Earliest Face Recognition Algorithms

Previous algorithms include algorithms based on engineering features, template matching algorithms, and subspace algorithms.

Engineering Features the tactic of using scope knowledge to pick and convert the foremost relevant variables from data when creating a predictive sample by using machine learning or statistical modeling. ***Template matching*** is deployed first to appear out the places of high correlation for the face and eyes templates. Subsequently, employing a mask derived from color segmentation and cleaned by texture filtering and various binary operations, the false and repeated hits are far from the template matching result. ***The sub-space algorithm*** treats the face image as a high-dimensional vector, and it conveys the vector into a low-dimensional space, while the low-dimensional vector obtained after projection achieves good differentiation of different people.

1.3.2 Synthetic Features and Classifier

This stage generally adopts the idea of artificial features and classifiers. The classifiers contain mature schemas, such as neural networks, support vector machines. The key is to design artificial features, which should be able to effectively distinguish between different people. Several features describing Faces have been used in face recognition problems, including HOG, SIFT, Gabor, LBP, etc.

A typical representative of them is the LBP (Local Binary Mode) feature, which is simple but effective. LBP features are very simple to calculate and partially solve light sensitivity problem, but still problems with parcels and expressions.

1.3.3 Deep learning

This stage is a method based on deep learning; many researchers have been trying to apply it in their own direction, which greatly encouraged the development of deep learning.

Convolutional neural networks show great strength in classifying images, and the convolutional kernel obtained through learning is much better than the artificially designed + distinct classifier diagram.

Face recognition researchers use (twisted neural networks) to find large numbers of facial images, then extract input features useful to distinguish between different people's faces, and to replace artificially designed features.

1.4 Face Recognition and Artificial Intelligence

Intelligent systems are being increasingly developed aiming to simulate our perception of various inputs (patterns) such as images, sounds...etc. Biometrics, in general, and facial recognition in particular are examples of popular applications for artificial intelligent systems. Intelligent face recognition system requires providing meaningful data and sufficient information during machine learning of a face. Facial recognition by machines can be Priceless and has various important applications in real life.

CHAPTER 2

2. PROBLEM STATEMENT AND PROPOSED SOLUTION

2.1 Technical Problems and Challenges of Face Recognition System

Often, images of our faces change naturally, because the facial recognition system undergoes a number of complexities during detection. Any facial recognition process can be described as (strong) or (weak) based on its perceived performance under a variety of challenging conditions. This type of problem is considered a separation problem. Selecting some of the images on our website and taking them as training photos and separating newly arrived images as test images in any given categories is a key step in the face recognition system. The title seems simple to the person, and is actually a really difficult task because of the limited memory of the program; moreover, problems with machine recognition are many.

We can extract these challenges in:

- **Find the same face (category similarity):** Different people may have the same look that at times it is not possible for an individual to view them.
- **Other methodical issues:** These issues can be caused by methods errors that are used in facial recognition, like input device deformation, background sensation, passive database, mistaken techniques etc. In addition of may be connection issues due to natural factors.

Accepting class diversity due to:

- **Head position:** Man is created in multiple forms that differ from one person to another. These differences are represented in the shape of the face Head movements, which can be defined by self-rotating rotation angles as the camera's flexible viewing angle can cause major changes in facial expressions and/or posture and produce variations of the subject's face, making automated face recognition everywhere difficult.
- **Lightness modes:** Lightness means a variation of illumination. Illumination changes can change the overall dimension of the illumination regenerate from an entity, as well as the dimming pattern and shading visual in the image. And the problem of facial recognition over light changes is widely seen as complexity in humans and algorithms. Difficulties caused by variable light conditions. It is establish the comparison between two conducted pictures of the same individual taken under various illumination is maximal than the difference between the images of two different individuals under the same illumination.
- **Facial transformation:** Some changes in facial expressions may be caused by a change in facial expressions caused by a variety of emotional states. Therefore, accurate and automatic recognition of different facial expressions is important in both emotional assessment and facial recognition. In particular, personal expressions are made up of large, expressive words, e.g. anger, disgust, fear, joy, sadness or surprise.
- **Facial Utilities:** Facial blindness, for example sunglasses, hats, scarves, beards, etc., can greatly affect the functioning of any facial recognition system.
- **The effects of aging:** One of the most important factors affecting the effectiveness of this technique is the individual's face is not unique. All things going for changing over time, so

over the years a person's appearance also changes which affects the facial recognition system.

2.2 Facial Recognition System criticisms:

Despite the great advantages of the technology, it is not without criticism, but it does not significantly affect its great benefits.

- Facial recognition technology threatens individual freedoms: This technology can collect and save images of individuals' faces and analyze their personal biometric data, in true timing, with else resources such as CCTV recordings, in order to identify citizens' identities regardless of whether they have committed a crime or not.
- Critics say facial recognition technology is being used in a legal vacuum, especially when the lines between the public sector and the private sector companies that use the technology become blurred. They suggest the need for government laws and policies to be remedied to keep pace with the rapid progress of artificial intelligence by providing facial recognition service.
- The European Public Data Protection Law Enactment Authority has established strict measures to protect personal data and handle sensitive private data such as medical records and children's records.
- In this regard, there are severe penalties. The organizations that process this data, whether private or public, must take appropriate security measures and ensure that they process the data in accordance with the necessary protection principles.

2.3 PROPOSED METHODOLOGY

We proposed method using appropriate algorithms which use face recognition standard implementation techniques using Local Binary Pattern Histogram algorithm (LBPH) and Additive Angular Margin Loss for Deep Face Recognition (ArcFace) Model.

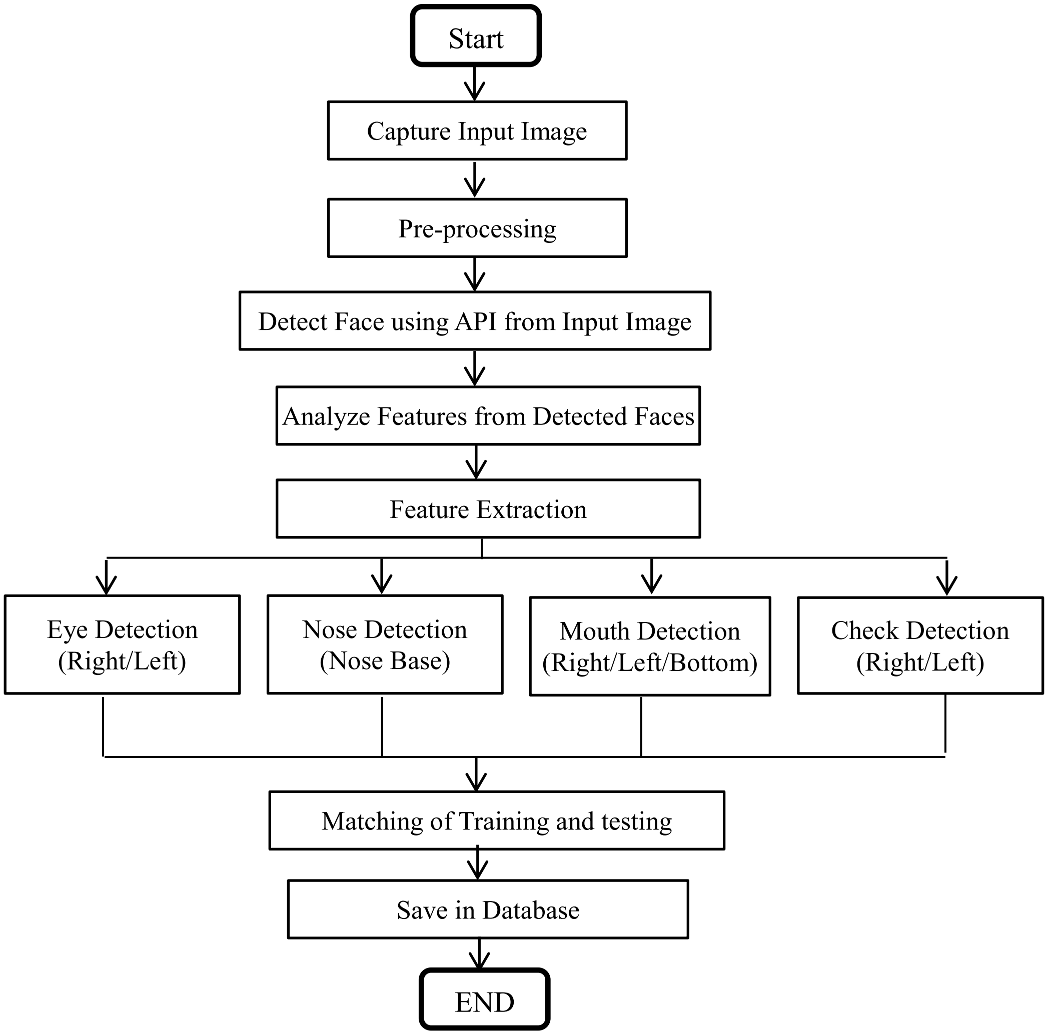

Start
Capture Input Image
Pre-processing
Detect Face using API from Input Image
Analyze Features from Detected Faces
Feature Extraction
Eye Detection (Right/Left)
Nose Detection (Nose Base)
Mouth Detection (Right/Left/Bottom)
Check Detection (Right/Left)
Matching of Training and testing
Save in Database
END

Figure 2.1: Flow Chart of proposed Model (Database Construction)

2.4 Proposed Solution

1. The new users should complete the registration; this process should take the face data of user.

2. The unique Identification (ID) was provided to the actual uses during the registration process.

3. The collected user face data and also the unique ID are going to be stored within the data base.

4. The registered merchants will have a tool to scan the face of the customer's. The user or payer just must show his/her face to the device which will scan the face. Users are wont to take the input of the face of the payer or customer. Then the user should enter his/her unique ID which his/her got during registration.

5. The Input of face from the device and therefore the unique ID which the user entered are going to be checked within the database.

6. If the face is matched with the info of the database and therefore the unique ID entered by the user are same as that of the unique ID of the face that are within the database then the transaction are processed to further process.
7. If the input from the device didn't match with any face data within the database or if the face got detected and therefore the unique ID that the user entered didn't match then the transaction are going to be failed and also the processes are starts from the first phase.

2.5 Local Binary Pattern Histogram Algorithm LBPH

It could be a face recognition algorithm that's very efficient texture factor prepared to recognize the front face and side face of any person we need to recognize him supported by or based on local binary factor which naming the pixels of a picture by Thresholding the neighborhood of every pixel and considers the outcome in a binary number format. LBPH improves the detection performance considerably on some datasets that is combined with histograms of oriented gradients (HOG) descriptor.

2.5.1 The LBPH operation

The primary LBPH computational step is make a moderate image to describe our source image in a proper way by using displaying the face features technique. The algorithm uses an idea of a slipping window established by the **radius parameters** and **neighbors parameters**.

2.5.2 Training the LBPH Algorithm

Training rules are wont to make sure the output decisions criteria and training algorithm will be used to get some input from the data to match the suitable output type. So, the algorithm and rules are used to simplify the method of learning. The proposed system uses the information gathered from the data to urge results. The precision and accuracy of the algorithm are verified by employing a test set of images.

2.5.3 System Block Diagram

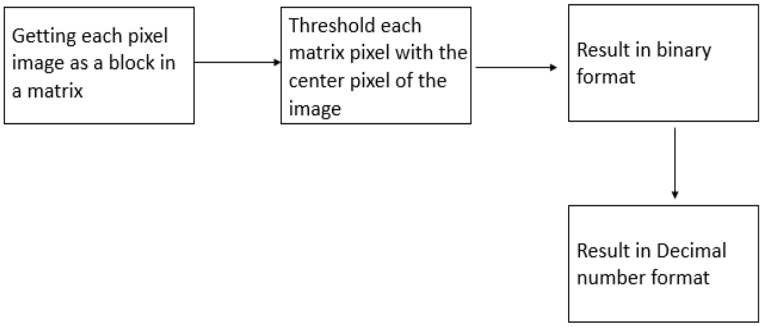

Getting each pixel image as a block in a matrix
Threshold each matrix pixel with the center pixel of the image
Result in binary format
Result in Decimal number format

Figure 2.2: Block Diagram of the Proposed Solution

The comprehensive proposed system block diagram is:

1) User inputs the image for recognizing purpose to the system through User Interface.
2) The aggregate proposed system is represented as a DeepNet block. Containing of Input processing and DeepNet processing layers.
3) Source image was processed and then feed it to the DeepNet layers for entity detection and classification.
4) The user interface will display the output image.

2.5.4 Data flow Diagram DFD

Facial recognition Data Flow Diagram (DFD) means a diagrammatic performance of the flow of data out of information system. Based on contactless payments it gives the general overview of the system without progression into deep detail.

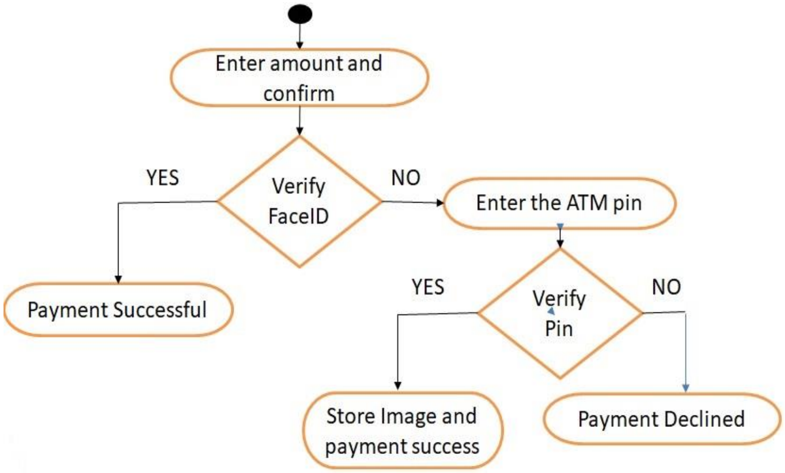


Figure 2.3: Secured Face Payment System Data Flow Diagram

The flow of the system is as follows:

1) First a customer signs up on his/her smart phone and enters payment amount and confirms.
2) Face verification well done by comparing input image with datasets, The System struggles at recognizing faces.
3) The payment process is successful, if face is successfully verified.
4) If face verification missing, will asking for pin code.
5) If entered pin code is verified, face image is captured for verification and payment is successful.
6) If not verified again, payment is rejected.

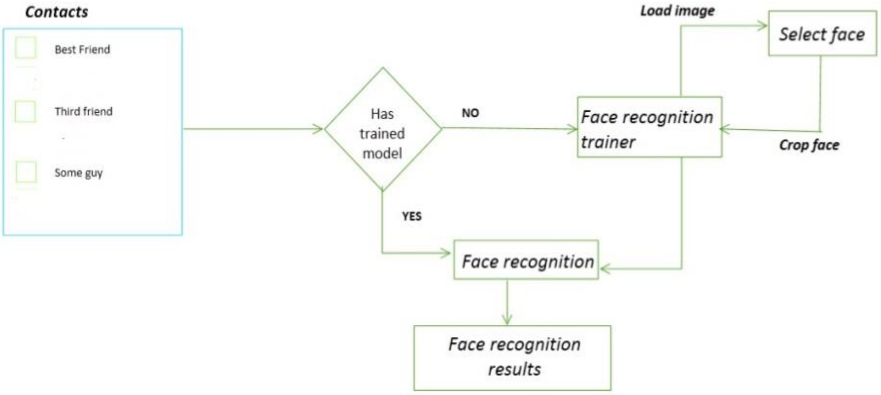


Figure 2.4: Phases in the Proposed Solution

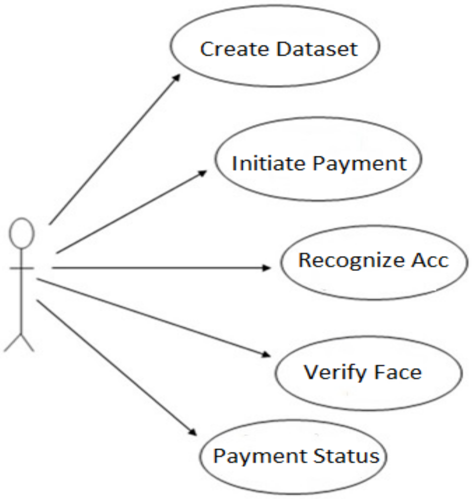


Figure 2.5: Use Case Diagram of Proposed Solution

This proposed system must be eligible to:

1. Establishing datasets for facial recognition and set credit card data.
2. Build payment method gate.
3. Set credit card data and recognizing customer account automatically.
4. Pick input customer's face image from at present then identify it with the valid account.
5. Verify user and making pay.
6. Capturing user's face image if face id doesn't verified.

2.6 ArcFace Model

This Machine Learning model based on the input images comparison of two faces to check if is that same person or not and should detect faces before recognizing faces; otherwise they can't extract facial features. It allows face detection when comparing facial features, face tracking, facial feature detection (gender, age), 3D face angle detection, and face comparison. Deep face Library introduces an overall analysis of face recognition models established by deep learning according to a new perspective that takes into account discrimination. These differences in face recognition models are most important as it can mark the value of proportion of these faces perfectly matched and faces un perfectly matched for a proven threshold.

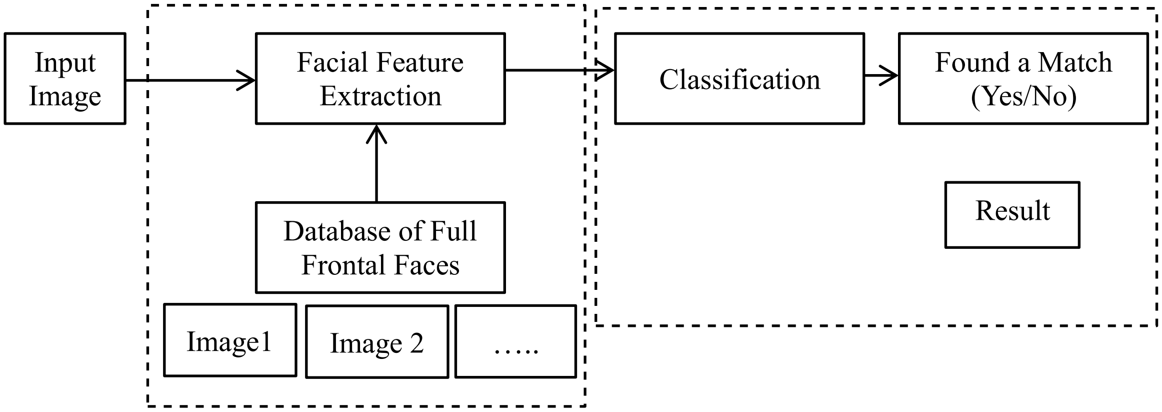


Figure 2.6: Principles of Face Recognition Models

Facial recognition is done extracting the face value A comparison of the facial attribute value obtains the similarity between *0.0 ~ 1.0* (that mean is closer the similarity to 1.0, and the more likely of two facial features data is one person).

The Arcface Model idea is loss function applied to consider the angle among the current feature and target weight representing *cos (θ+ m)* directly maximize decision boundary in angular (arc) space based on normalized weights and features.

ArcFace in the normalized face features lies directly optimizes the geodesic distance margin.

CHAPTER 3

3. LITERATURE REVIEW

Facial Recognition assuming it's a modern way of recognizing an individual’s face over technology it is possible to build a system using measurements related to the human body to map facial features from a face image or video streaming. It can compare the input data and checks the details of important information that stored in a database of known faces to find a perfect match. Most recent efforts have been made in different types of biometrics such as fingerprint recognition. Now because of increasing quality of pictures capturing devices by users of electronic devices (Such as laptops, tablets, and smart phones) has enabled the extension of comparatively that accessible to everyone software and apps offering device-based facial recognition developments constitute a form of biometric technology relying on measurements of human characteristics this technology is used in security [1][2][3][9][10]. Researches [4][5][6][11] shows the development of a theoretical and algorithmic basis that Computer vision concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of image and concerned with the theory behind artificial systems that extract information from input images. Now there are many standards implementation of face recognition techniques of Deep Face Recognition [18][19][20]. Other researches [7][8] used Deep Learning Convolution Neural Networks used to train model that extract deep features from images of faces in addition to that some of researches using different machine learning classifiers. Research proposed a general design of payment system by adopting some methods. The features of Face recognition payment systems is offer services to investigate resistance to use, giving businesses information to refer to. It can use a gateway for E-payment by face recognition system instead of password based transaction and verify a person is proven to successful transaction [12][13][14][15][16][17].

CHAPTER 4

4. IMPLEMENTATION

4.1 Implementation Platform

4.1.1 Hardware Component

- CPU: Intel (R) Core(TM) i3
- RAM: 8 GB
- GPU: NVIDIA Tesla K80

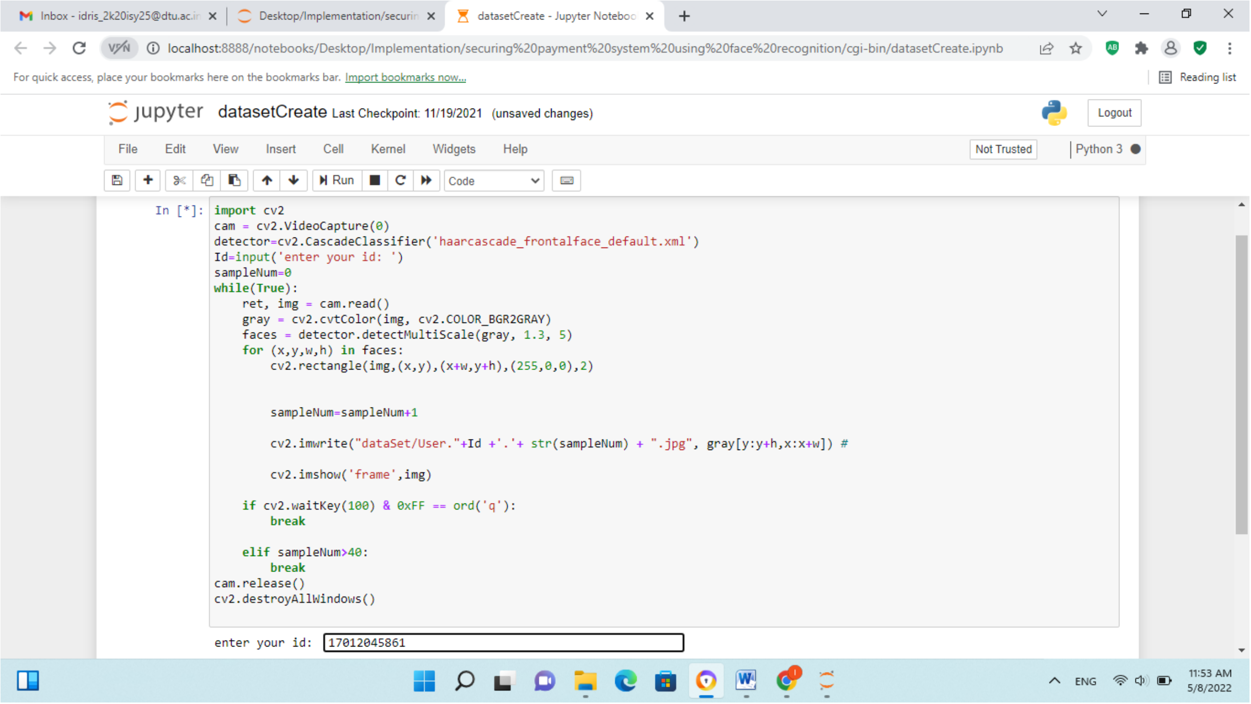
4.1.2 Software

- Operating System: Ubuntu (14) (64bit), Windows 11 (64 bit)
- Programming Languages: C, Python, CSS, php5, html 5
- Server: WAMP Server (Windows, Apache, MySQL, and PHP)

4.2 LBPH Implementation Details

4.2.1 Dataset Collection

- This process it is compulsory steps of all the system.
- Creating account details by the user over the credit card details and building a dataset of account's face.
- The camera constantly takes 40 images in a spread of just seconds.
- This images located in database which is connected with the user's ID.
- The images of faces taken are first converted to gray-scale, and then contour mapping is done on them.
- This enhances the ability of the applied algorithm to recognize facial patterns more efficiently.
- This enables the face recognition algorithm to quickly recognize a face all in a fraction of a second.
- The Haarcascade Classifier does the main task of creating the dataset.
- This classifier has an exceptional ability to pick up images from real time footage using the webcam.



```
import cv2
cam = cv2.VideoCapture(0)
detector=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
Id=input('enter your id: ')
sampleNum=0
while(True):
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)


        sampleNum=sampleNum+1

        cv2.imwrite("dataSet/User."+Id +'.'+ str(sampleNum) + ".jpg", gray[y:y+h,x:x+w]) #

        cv2.imshow('frame',img)

    if cv2.waitKey(100) & 0xFF == ord('q'):
        break

    elif sampleNum>40:
        break
cam.release()
cv2.destroyAllWindows()
```

```
enter your id:  17012045861
```

Figure 4.1: Create Dataset by Input Face Image with ID

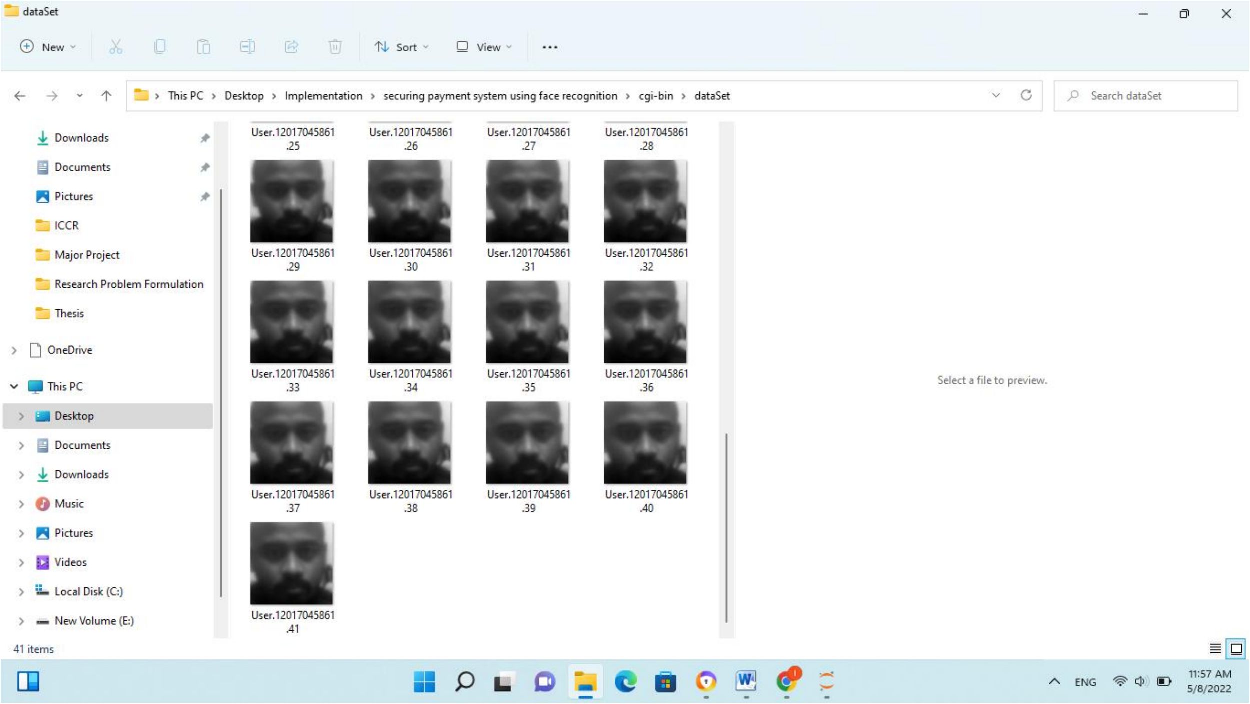


Figure 4.2: Create 41 face images including IDs into dataset

4.2.2 Training the Datasets

Face recognition technology can be achieved with the help of a learning concept of training and then testing the model with a given dataset.

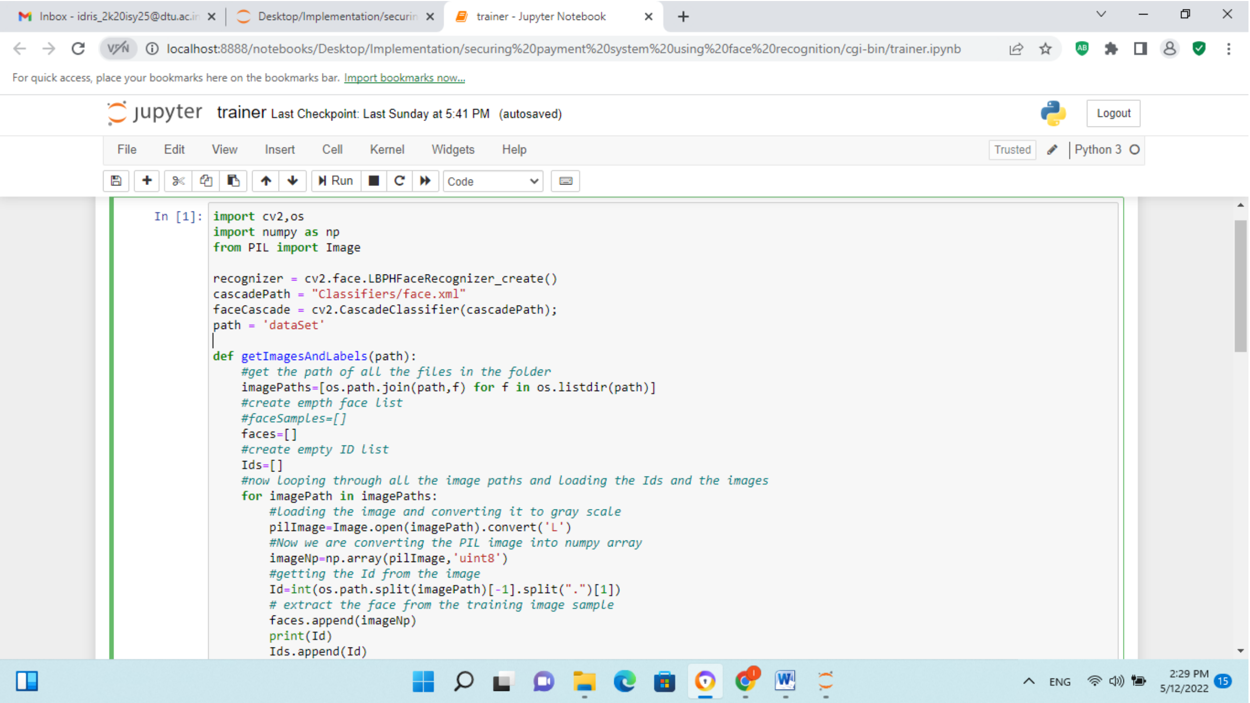
When the dataset is created, it has to be trained using their related IDs which have to match it to the user's card data. All this is stored in a separate file named (trainer.yml).

4.2.3 User Interface

We can set up for launching any web interface for the project it can be done by installing web server such as apache2, WAMP Sever, PHPv5 and transferring the User Interface UI files into the instance. The code files are then saved in the /var/www/html folder so as to be accessible by the server.

4.2.4 Testing on Training Dataset

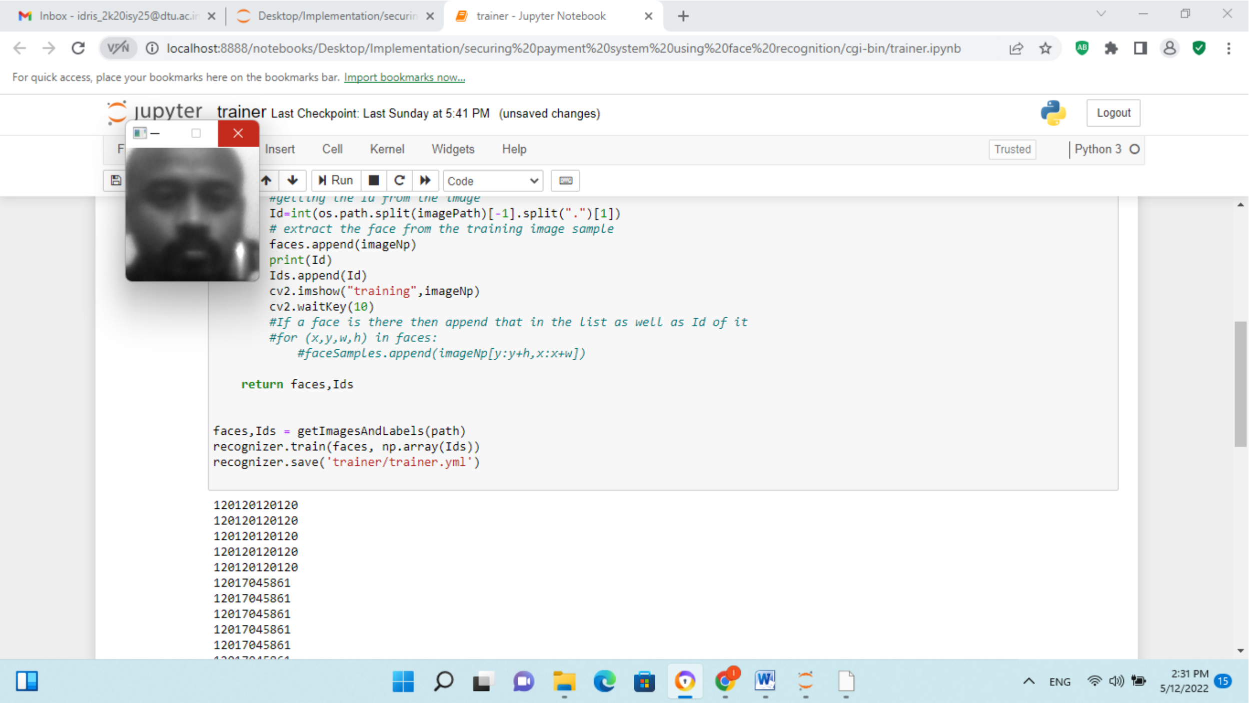
Images from the training dataset were tested on the network and the results were as follows:


Inbox - idris_2k20isy25@dtu.ac.in
Desktop/Implementation/securin
trainer - Jupyter Notebook
localhost:8888/notebooks/Desktop/Implementation/securing%20payment%20system%20using%20face%20recognition/cgi-bin/trainer.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter trainer Last Checkpoint: Last Sunday at 5:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
2:29 PM
5/12/2022

```
In [1]: import cv2,os
        import numpy as np
        from PIL import Image

        recognizer = cv2.face.LBPHFaceRecognizer_create()
        cascadePath = "Classifiers/face.xml"
        faceCascade = cv2.CascadeClassifier(cascadePath);
        path = 'dataSet'

        def getImagesAndLabels(path):
            #get the path of all the files in the folder
            imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
            #create empth face list
            #faceSamples=[]
            faces=[]
            #create empty ID list
            Ids=[]
            #now looping through all the image paths and loading the Ids and the images
            for imagePath in imagePaths:
                #loading the image and converting it to gray scale
                pilImage=Image.open(imagePath).convert('L')
                #Now we are converting the PIL image into numpy array
                imageNp=np.array(pilImage,'uint8')
                #getting the Id from the image
                Id=int(os.path.split(imagePath)[-1].split(".")[1])
                # extract the face from the training image sample
                faces.append(imageNp)
                print(Id)
                Ids.append(Id)
```

Figure 4.3: Training the dataset

In this figure we Training dataset by getting the path of all the files and loading images with IDs and converting it in gray-scale mode.

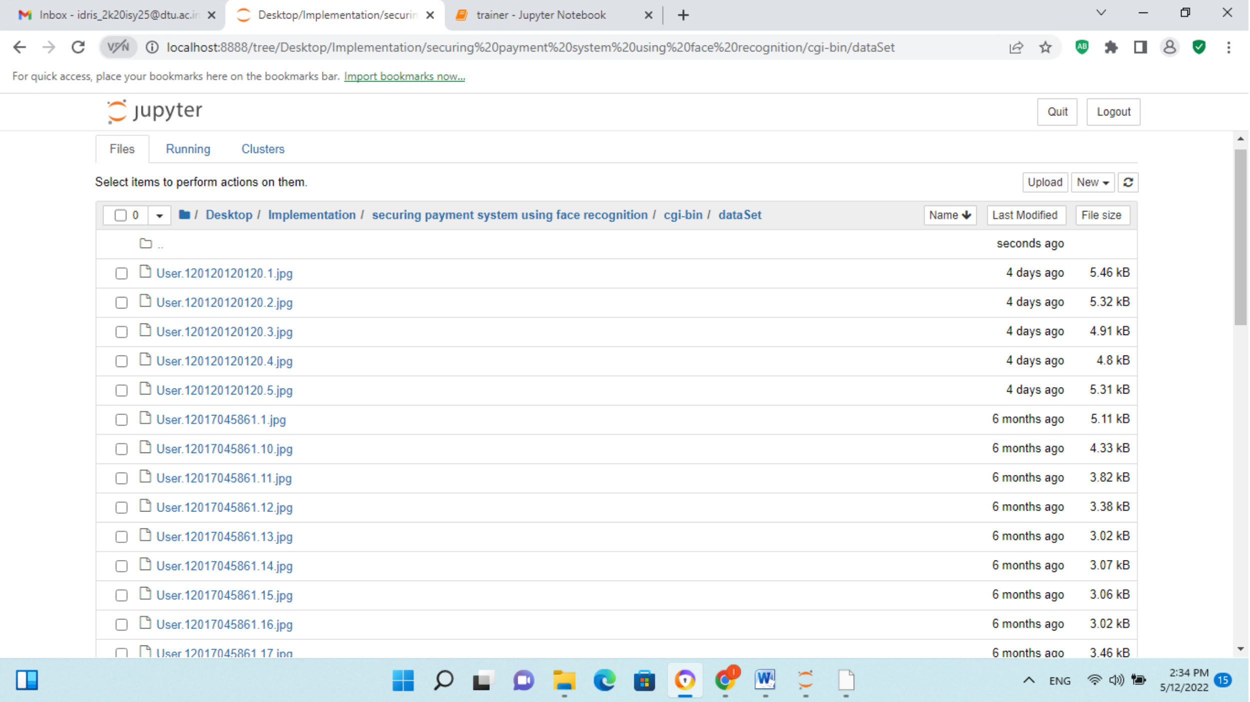

Inbox - idris_2k20isy25@dtu.ac.in
Desktop/Implementation/securin
trainer - Jupyter Notebook
localhost:8888/notebooks/Desktop/Implementation/securing%20payment%20system%20using%20face%20recognition/cgi-bin/trainer.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter trainer Last Checkpoint: Last Sunday at 5:41 PM (unsaved changes)
Logout
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
ENG
2:31 PM
5/12/2022

```
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        print(Id)
        Ids.append(Id)
        cv2.imshow("training",imageNp)
        cv2.waitKey(10)
        #If a face is there then append that in the list as well as Id of it
        #for (x,y,w,h) in faces:
            #faceSamples.append(imageNp[y:y+h,x:x+w])

    return faces,Ids


faces,Ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(Ids))
recognizer.save('trainer/trainer.yml')
```

```
120120120120
120120120120
120120120120
120120120120
120120120120
12017045861
12017045861
12017045861
12017045861
12017045861
```

Figure 4.4: The result of our training the dataset

This figure shows the result of our training dataset by getting the Images with IDs.


Inbox - idris_2k20isy25@dtu.ac.in
Desktop/Implementation/securin
trainer - Jupyter Notebook
localhost:8888/tree/Desktop/Implementation/securing%20payment%20system%20using%20face%20recognition/cgi-bin/dataSet
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter
Quit
Logout
Files
Running
Clusters
Upload
New
ENG
2:34 PM
5/12/2022

Select items to perform actions on them.

/ Desktop / Implementation / securing payment system using face recognition / cgi-bin / dataSet

Name	Last Modified	File size
..	seconds ago	
User.120120120120.1.jpg	4 days ago	5.46 kB
User.120120120120.2.jpg	4 days ago	5.32 kB
User.120120120120.3.jpg	4 days ago	4.91 kB
User.120120120120.4.jpg	4 days ago	4.8 kB
User.120120120120.5.jpg	4 days ago	5.31 kB
User.12017045861.1.jpg	6 months ago	5.11 kB
User.12017045861.10.jpg	6 months ago	4.33 kB
User.12017045861.11.jpg	6 months ago	3.82 kB
User.12017045861.12.jpg	6 months ago	3.38 kB
User.12017045861.13.jpg	6 months ago	3.02 kB
User.12017045861.14.jpg	6 months ago	3.07 kB
User.12017045861.15.jpg	6 months ago	3.06 kB
User.12017045861.16.jpg	6 months ago	3.02 kB
User.12017045861.17.jpg	6 months ago	3.46 kB

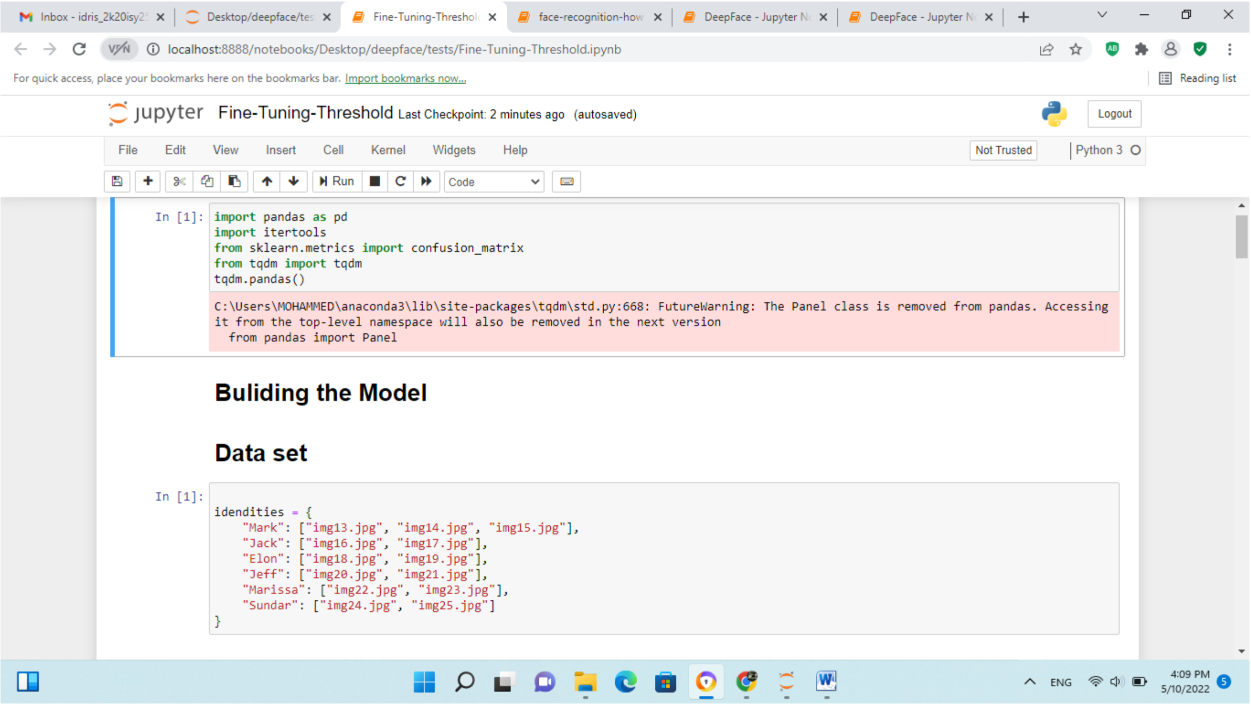
Figure 4.5: Images from the test dataset

This figure shows the image faces resulting from Creating Dataset process. These images can be stored in database and query it when we need it.

4.3 ArcFace Implementation Details

This Face recognition model is one of the regular convolutional neural networks models. It represents face Images as vectors. We find the distance between these two vectors to compare two faces. Finally, we classify two faces as same person whose distance is less than a threshold value.

The most important thing is that how to determine the threshold. In this notebook, we will find the best split point for a threshold.



localhost:8888/notebooks/Desktop/deepface/tests/Fine-Tuning-Threshold.ipynb
jupyter Fine-Tuning-Threshold Last Checkpoint: 2 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code

```
In [1]: import pandas as pd
        import itertools
        from sklearn.metrics import confusion_matrix
        from tqdm import tqdm
        tqdm.pandas()
```

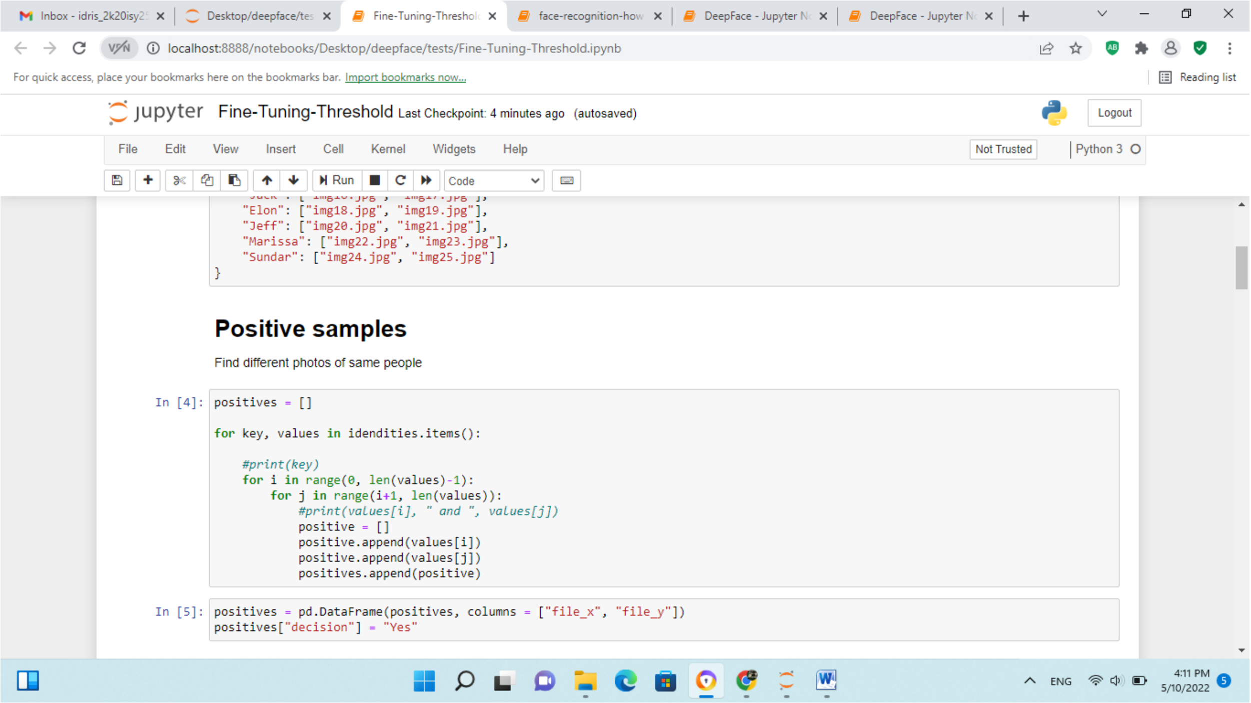
```
C:\Users\MOHAMMED\anaconda3\lib\site-packages\tqdm\std.py:668: FutureWarning: The Panel class is removed from pandas. Accessing
it from the top-level namespace will also be removed in the next version
  from pandas import Panel
```

Buliding the Model

Data set

```
In [1]:
idendities = {
    "Mark": ["img13.jpg", "img14.jpg", "img15.jpg"],
    "Jack": ["img16.jpg", "img17.jpg"],
    "Elon": ["img18.jpg", "img19.jpg"],
    "Jeff": ["img20.jpg", "img21.jpg"],
    "Marissa": ["img22.jpg", "img23.jpg"],
    "Sundar": ["img24.jpg", "img25.jpg"]
}
```

Figure 4.6: Create a dataset of Faces to build the ArcFace model


Inbox - idris_2k20isy2
Desktop/deepface/tes
Fine-Tuning-Threshol
face-recognition-how
DeepFace - Jupyter N
DeepFace - Jupyter N
localhost:8888/notebooks/Desktop/deepface/tests/Fine-Tuning-Threshold.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Reading list
jupyter
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
ENG
4:11 PM
5/10/2022

Fine-Tuning-Threshold Last Checkpoint: 4 minutes ago (autosaved)

```
    "Jack": ["img16.jpg", "img17.jpg"],
    "Elon": ["img18.jpg", "img19.jpg"],
    "Jeff": ["img20.jpg", "img21.jpg"],
    "Marissa": ["img22.jpg", "img23.jpg"],
    "Sundar": ["img24.jpg", "img25.jpg"]
}
```

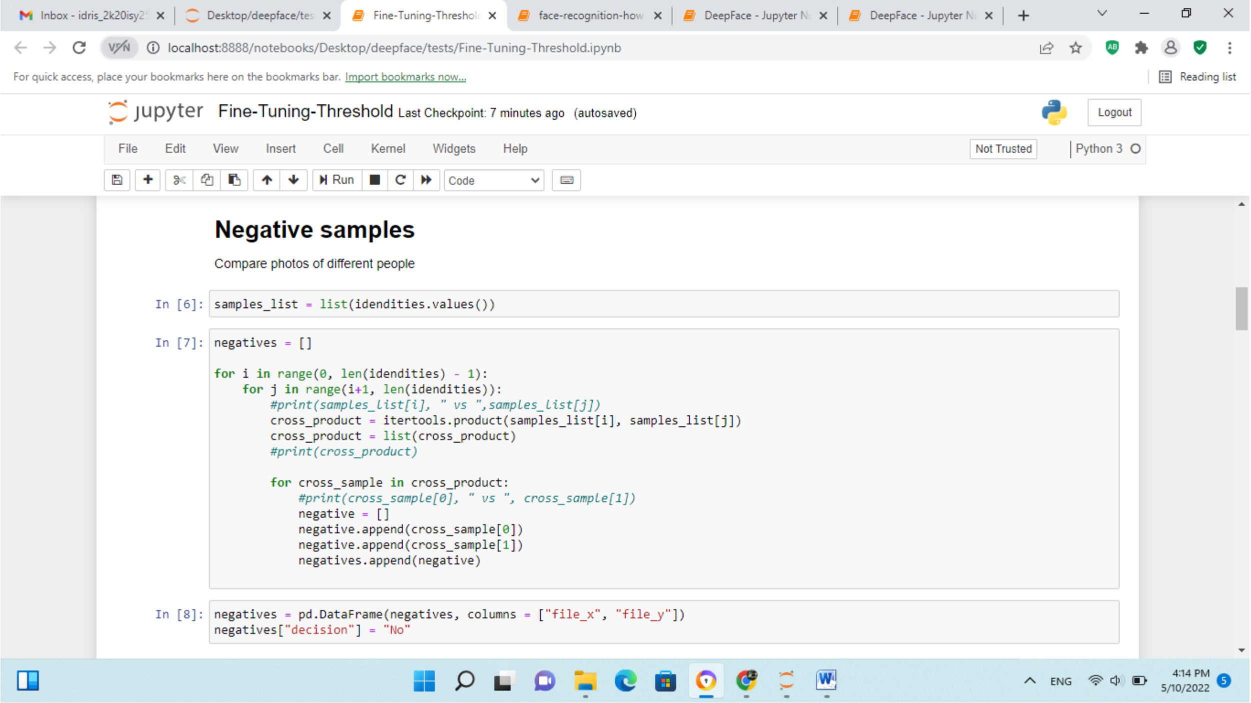
Positive samples

Find different photos of same people

```
In [4]: positives = []

        for key, values in idendities.items():

            #print(key)
            for i in range(0, len(values)-1):
                for j in range(i+1, len(values)):
                    #print(values[i], " and ", values[j])
                    positive = []
                    positive.append(values[i])
                    positive.append(values[j])
                    positives.append(positive)
```

```
In [5]: positives = pd.DataFrame(positives, columns = ["file_x", "file_y"])
        positives["decision"] = "Yes"
```

Figure 4.7: Find different photos of same people as A Positive Samples


Inbox - idris_2k20isy2
Desktop/deepface/tes
Fine-Tuning-Threshol
face-recognition-how
DeepFace - Jupyter N
DeepFace - Jupyter N
localhost:8888/notebooks/Desktop/deepface/tests/Fine-Tuning-Threshold.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Reading list
jupyter
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
ENG
4:14 PM
5/10/2022

Fine-Tuning-Threshold Last Checkpoint: 7 minutes ago (autosaved)

Negative samples

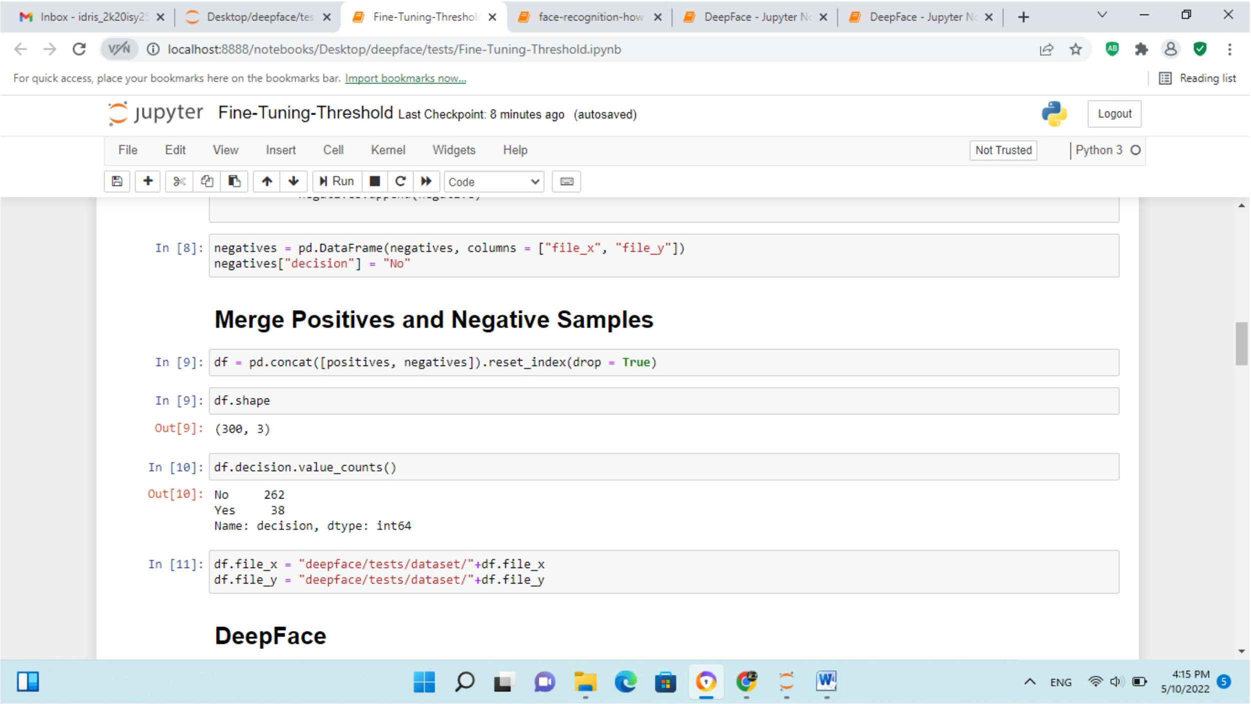
Compare photos of different people

```
In [6]: samples_list = list(idendities.values())
```

```
In [7]: negatives = []

        for i in range(0, len(idendities) - 1):
            for j in range(i+1, len(idendities)):
                #print(samples_list[i], " vs ",samples_list[j])
                cross_product = itertools.product(samples_list[i], samples_list[j])
                cross_product = list(cross_product)
                #print(cross_product)

                for cross_sample in cross_product:
                    #print(cross_sample[0], " vs ", cross_sample[1])
                    negative = []
                    negative.append(cross_sample[0])
                    negative.append(cross_sample[1])
                    negatives.append(negative)
```

```
In [8]: negatives = pd.DataFrame(negatives, columns = ["file_x", "file_y"])
        negatives["decision"] = "No"
```

Figure 4.8: Find different photos of different people as A Negative Samples


localhost:8888/notebooks/Desktop/deepface/tests/Fine-Tuning-Threshold.ipynb
jupyter Fine-Tuning-Threshold Last Checkpoint: 8 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Logout
Run
Code
4:15 PM
5/10/2022

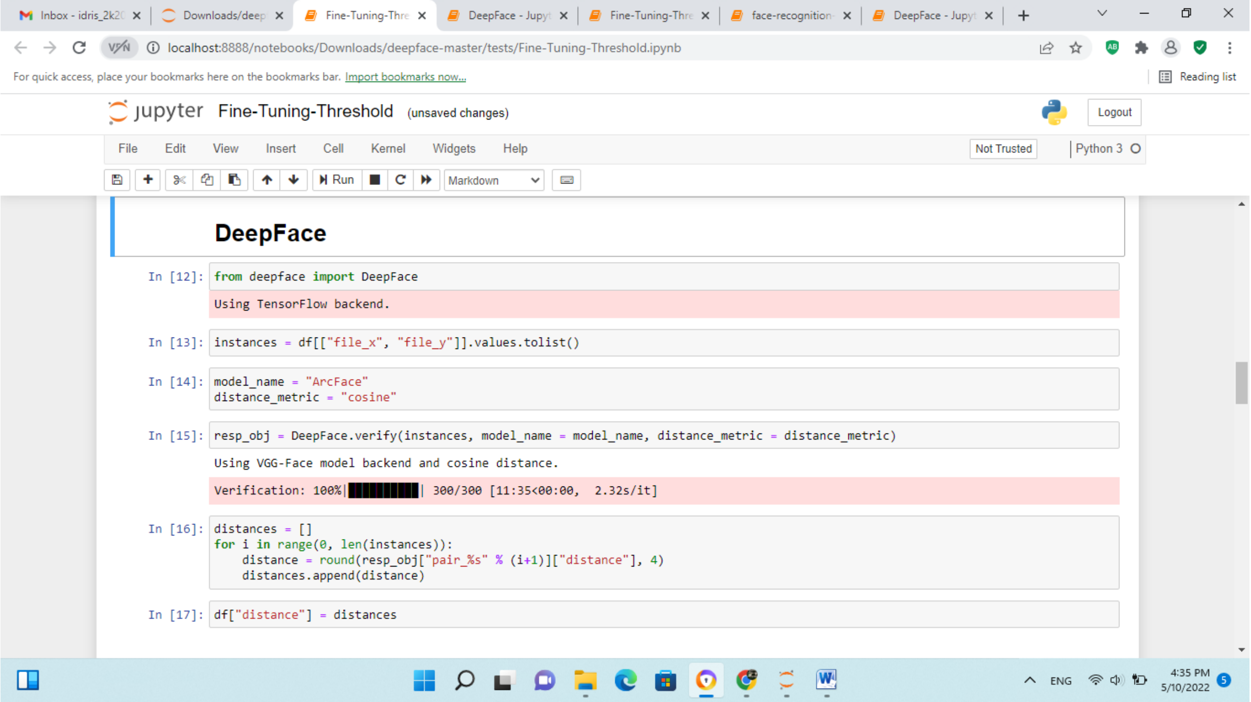
```
In [8]:  negatives = pd.DataFrame(negatives, columns = ["file_x", "file_y"])
         negatives["decision"] = "No"
```

Merge Positives and Negative Samples

```
In [9]:  df = pd.concat([positives, negatives]).reset_index(drop = True)

In [9]:  df.shape

Out[9]:  (300, 3)

In [10]: df.decision.value_counts()

Out[10]: No     262
         Yes     38
         Name: decision, dtype: int64

In [11]: df.file_x = "deepface/tests/dataset/"+df.file_x
         df.file_y = "deepface/tests/dataset/"+df.file_y
```

DeepFace

Figure 4.9: Merge the Positive samples and Negative Samples

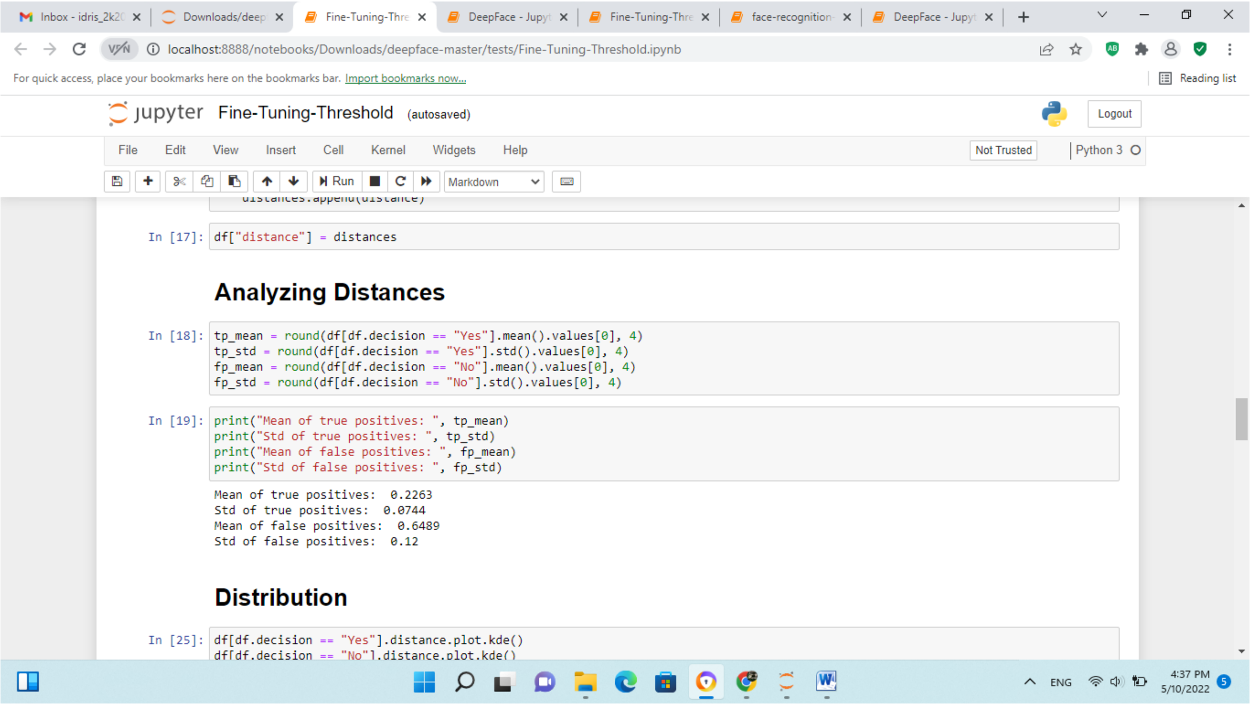

localhost:8888/notebooks/Downloads/deepface-master/tests/Fine-Tuning-Threshold.ipynb
jupyter Fine-Tuning-Threshold (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Logout
Run
Markdown
4:35 PM
5/10/2022

DeepFace

```
In [12]: from deepface import DeepFace

         Using TensorFlow backend.

In [13]: instances = df[["file_x", "file_y"]].values.tolist()

In [14]: model_name = "ArcFace"
         distance_metric = "cosine"

In [15]: resp_obj = DeepFace.verify(instances, model_name = model_name, distance_metric = distance_metric)

         Using VGG-Face model backend and cosine distance.

         Verification: 100%|██████████| 300/300 [11:35<00:00,  2.32s/it]

In [16]: distances = []
         for i in range(0, len(instances)):
             distance = round(resp_obj["pair_%s" % (i+1)]["distance"], 4)
             distances.append(distance)

In [17]: df["distance"] = distances
```

Figure 4.10: Running ArcFace in DeepFace

Here we run verify the face to find an identity in a database. All we have to do is just apply the model name to (ArcFace). We can detect an identity in a database quickly by Deepface in order to stores the representations of database items in advance.


Inbox - idris_2k2
Downloads/deep
Fine-Tuning-Thre
DeepFace - Jupy
Fine-Tuning-Thre
face-recognition
DeepFace - Jupy
localhost:8888/notebooks/Downloads/deepface-master/tests/Fine-Tuning-Threshold.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Reading list
jupyter Fine-Tuning-Threshold (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
ENG
4:37 PM
5/10/2022

```
    distances.append(distance)
```

```
In [17]: df["distance"] = distances
```

Analyzing Distances

```
In [18]: tp_mean = round(df[df.decision == "Yes"].mean().values[0], 4)
         tp_std = round(df[df.decision == "Yes"].std().values[0], 4)
         fp_mean = round(df[df.decision == "No"].mean().values[0], 4)
         fp_std = round(df[df.decision == "No"].std().values[0], 4)
```

```
In [19]: print("Mean of true positives: ", tp_mean)
         print("Std of true positives: ", tp_std)
         print("Mean of false positives: ", fp_mean)
         print("Std of false positives: ", fp_std)

         Mean of true positives:  0.2263
         Std of true positives:  0.0744
         Mean of false positives:  0.6489
         Std of false positives:  0.12
```

Distribution

```
In [25]: df[df.decision == "Yes"].distance.plot.kde()
         df[df.decision == "No"].distance.plot.kde()
```

Figure 4.11: Analyzing Distances

Face recognition requires O(n) time complexity and this becomes problematic for millions level data

CHAPTER 5

5. Experiments and Results

5.1 LBPH Algorithms

The prediction percentage and the accuracy of the bounding boxes in the results depend on the:

1) **Batch size:** is the number of images that are trained per batch in one iteration of training.

2) **Learning rate:** is the training parameter that controls the size of weight and bias changes during learning.

3) **Number of training iterations:** Learning Rate Number of Iterations is the number of training iterations after which the network is optimally trained.

IOU: abbreviation of **I**ntersection **o**ver **U**nion it is an evaluation metric accustomed measure the accuracy of an object detector on a specific dataset. Within the numerator we compute the area of overlap between the predicted bounding box and also the ground-truth bounding box. The denominator is that the area of union or more simply, the area encompassed by both the expected bounding box and also the ground-truth bounding box. Will get it by dividing the area of overlap by the area of union yields our final score the Intersection over Union.

5.1.1 Establishing Optimal parameters

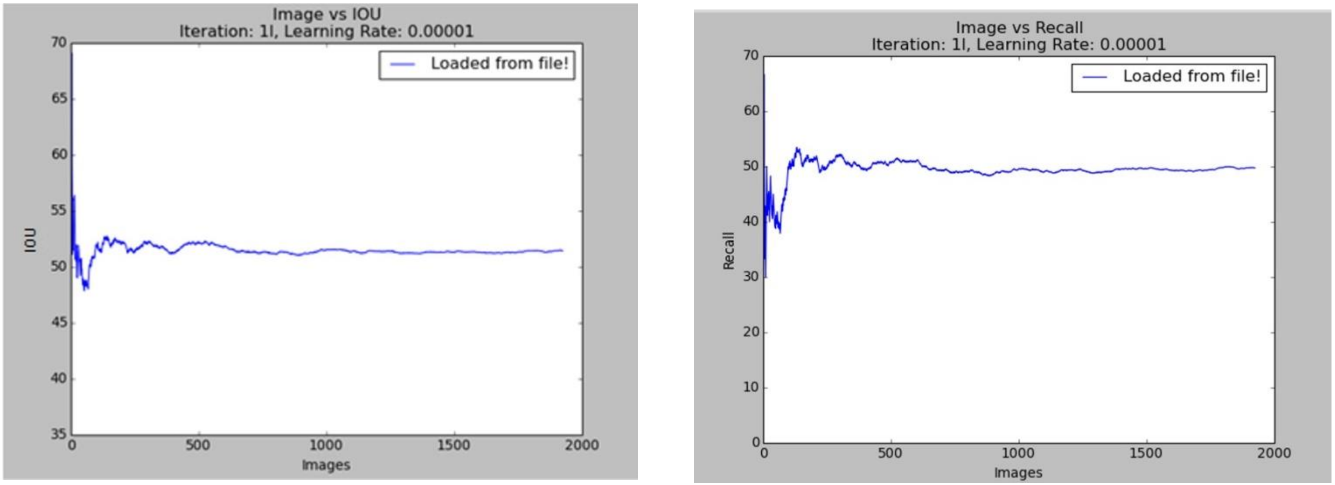
This section deals with experimenting with various training parameters are carried out to determine on an optimal set of parameters.

Batch Size

- Batch sizes were set to 64, 8 and 1 and experiments were carried out.
- Batch sizes of 64 and 8 took a long time to train and did not produce bounding boxes for object detection.
- Batch size 1 proved optimal with a fast training speed and also efficient bounding box predictions when tested

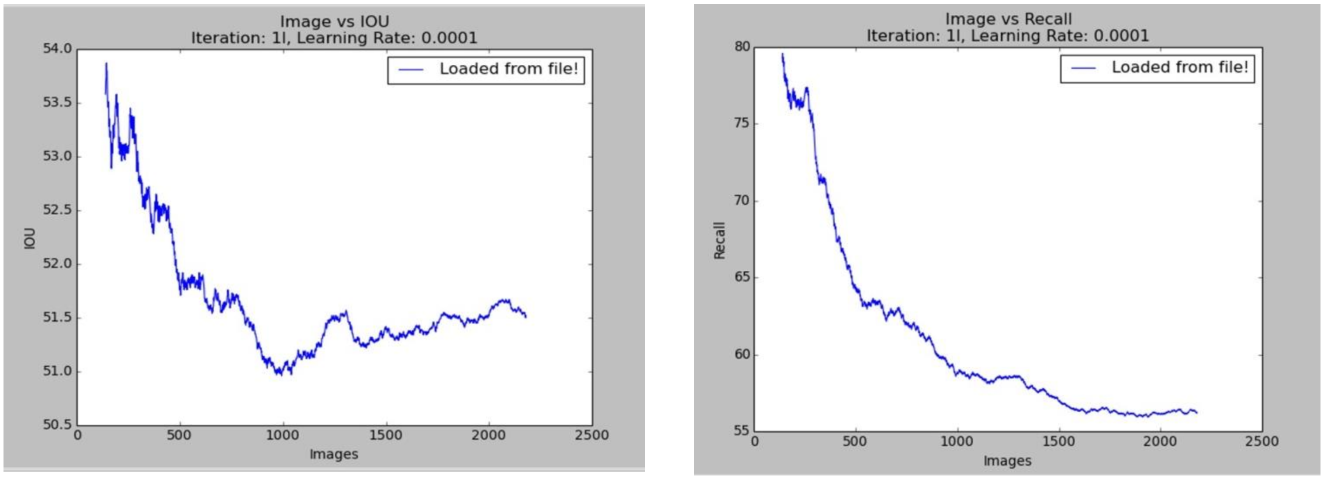
Learning Rate

1. Learning Rate (0.00001)



IOU per Image **Recall per Image**

Figure 5.1: IOU and Recall per Image for 0.00001 learning rate



IOU per Image **Recall per Image**

Figure 5.2: IOU and Recall per Image for 0.0001 learning rate

5.1.2 Optimal Parameters

After conducting the above experiments with respect to varying training parameters like Batch Size, number of training iterations and Learning Rate, the optimal values were found to be as follows:

- **Batch Size:** 1

When every iteration trains one image, it was found that the features of the image were learnt better.

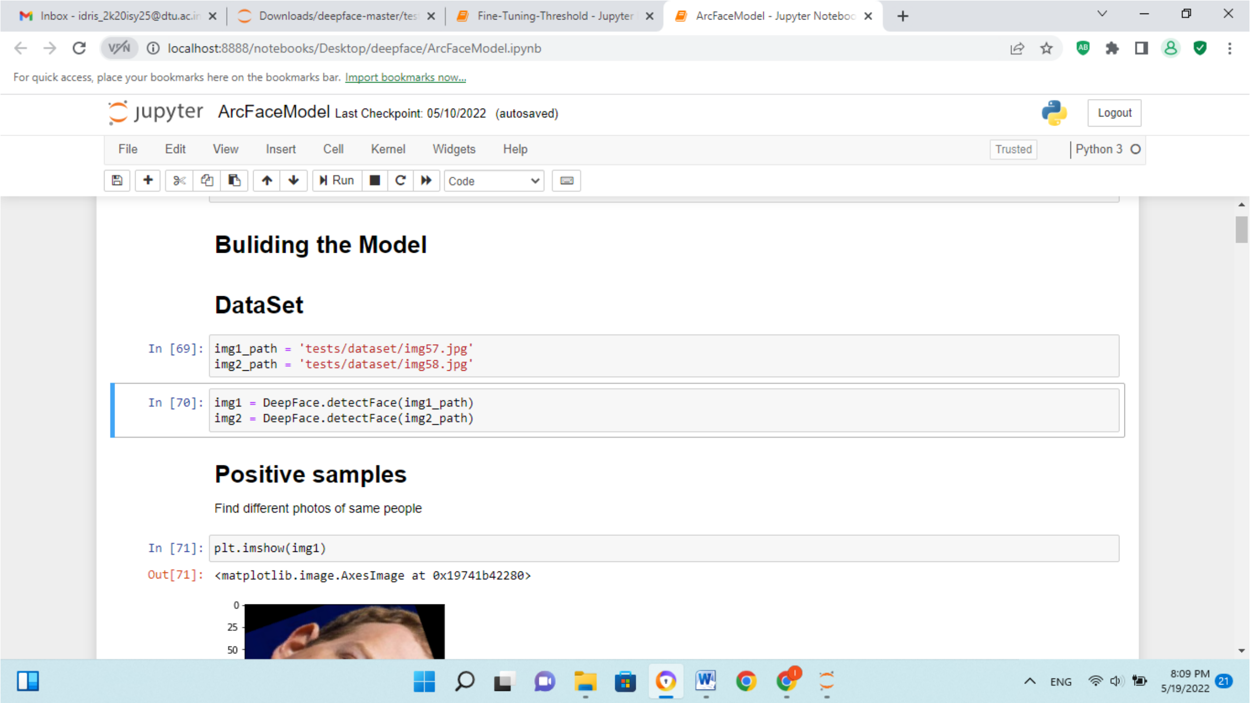
- **Learning Rate:** 0.00001

With Learning Rate below 0.00001, it had been found that the network was unable to detect objects in any respect. Learning Rate of 0.0001 was detected objects, but with lesser accuracy in terms of classification. Hence, the optimal learning rate for better detection and classification is found to be 0.00001

- **Number of training iterations:** 800000

For training below 800000 iterations, the network was found to predict fewer bounding boxes with less accuracy. For training for iterations from 900000 to 1000000, the network was found to be over fit, performing well only on the training images in dataset. Hence, the optimal number of training iterations was concluded to be as 800000.

5.2 ArcFace Experiments


Inbox - idris_2k20isy25@dtu.ac.in
Downloads/deepface-master/tes
Fine-Tuning-Threshold - Jupyter
ArcFaceModel - Jupyter Noteboo
localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter ArcFaceModel Last Checkpoint: 05/10/2022 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
0
25
50
ENG
8:09 PM
5/19/2022
21

Buliding the Model

DataSet

```
In [69]: img1_path = 'tests/dataset/img57.jpg'
         img2_path = 'tests/dataset/img58.jpg'
```

```
In [70]: img1 = DeepFace.detectFace(img1_path)
         img2 = DeepFace.detectFace(img2_path)
```

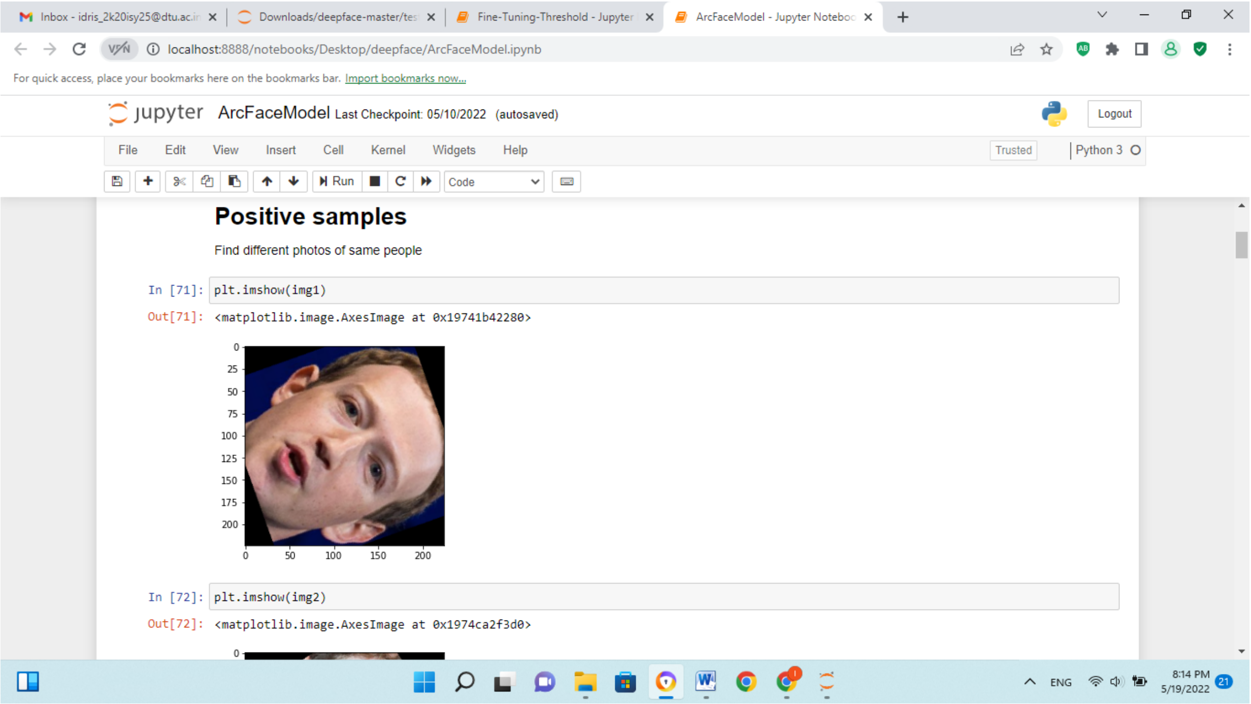
Positive samples

Find different photos of same people

```
In [71]: plt.imshow(img1)
Out[71]: <matplotlib.image.AxesImage at 0x19741b42280>
```

Figure 5.3: Building ArcFace Model for our experiment

We declare the path of two input images for this experiment and using deep face detect face function of both images.


Inbox - idris_2k20isy25@dtu.ac.in
Downloads/deepface-master/tes
Fine-Tuning-Threshold - Jupyter
ArcFaceModel - Jupyter Noteboo
localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter
ArcFaceModel Last Checkpoint: 05/10/2022 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
0
25
50
75
100
125
150
175
200
0
50
100
150
200
ENG
8:14 PM
5/19/2022
21

Positive samples

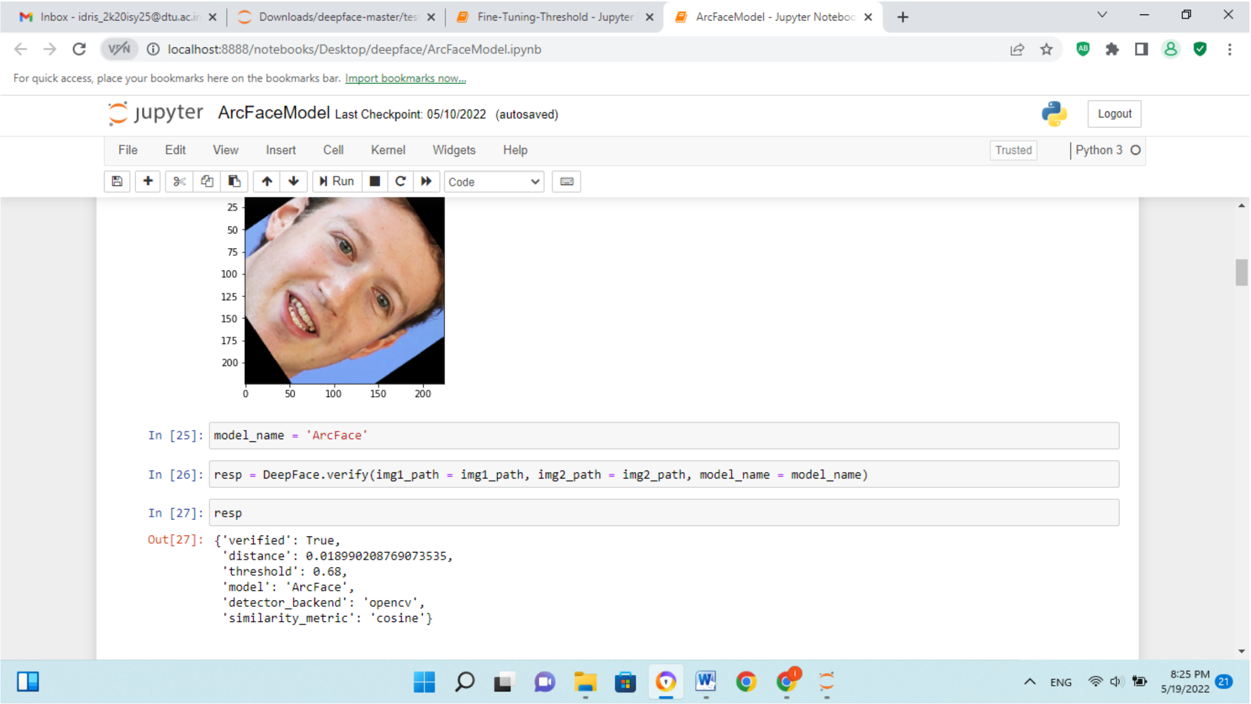
Find different photos of same people

```
In [71]: plt.imshow(img1)
Out[71]: <matplotlib.image.AxesImage at 0x19741b42280>
```

```
In [72]: plt.imshow(img2)
Out[72]: <matplotlib.image.AxesImage at 0x1974ca2f3d0>
```

Figure 5.4: show the images samples in ArcFace Model

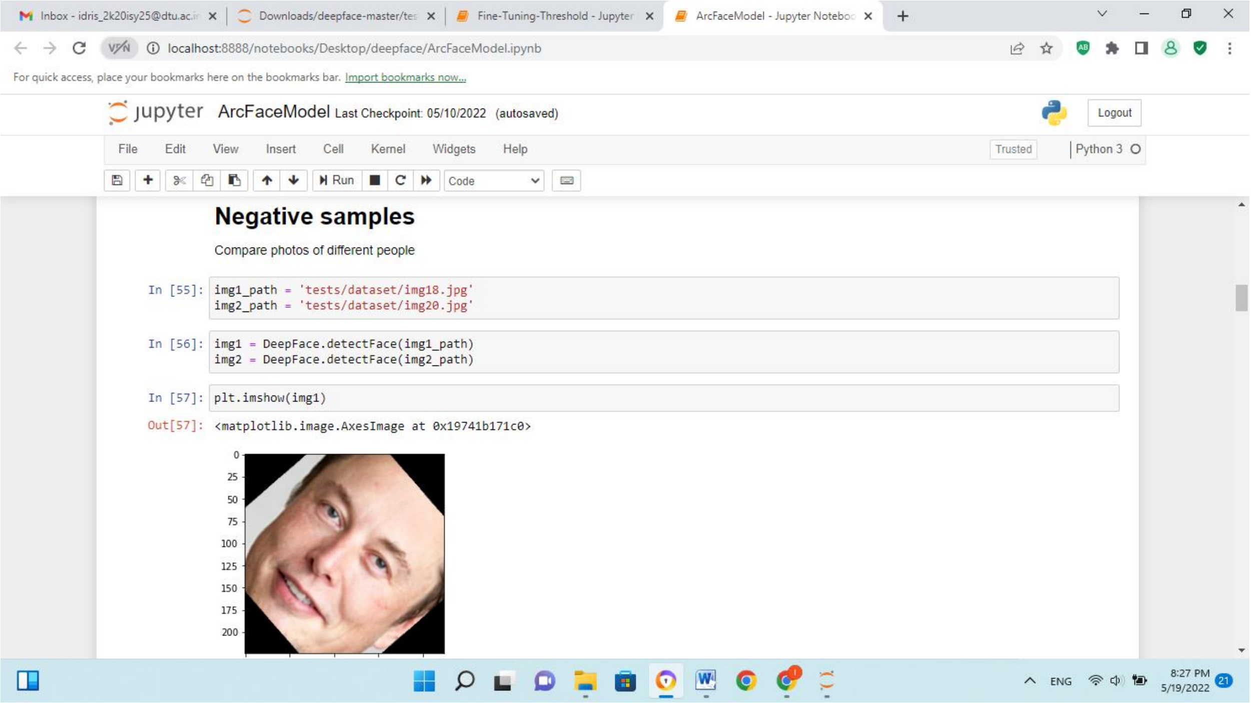
Here we test our model to find different images of same person (Positive Sample) we use the face detect function to find face in that image.


Inbox - idris_2k20isy25@dtu.ac.in
Downloads/deepface-master/tes
Fine-Tuning-Threshold - Jupyter
ArcFaceModel - Jupyter Noteboo
localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter
ArcFaceModel Last Checkpoint: 05/10/2022 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
25
50
75
100
125
150
175
200
0
50
100
150
200
ENG
8:25 PM
5/19/2022
21

```
In [25]: model_name = 'ArcFace'

In [26]: resp = DeepFace.verify(img1_path = img1_path, img2_path = img2_path, model_name = model_name)

In [27]: resp
Out[27]: {'verified': True,
          'distance': 0.018990208769073535,
          'threshold': 0.68,
          'model': 'ArcFace',
          'detector_backend': 'opencv',
          'similarity_metric': 'cosine'}
```

Figure 5.5: verify images in Positive Sample

The result of verified samples image is True (same person).


Inbox - idris_2k20isy25@dtu.ac.in
Downloads/deepface-master/tes
Fine-Tuning-Threshold - Jupyter
ArcFaceModel - Jupyter Notebook
localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter ArcFaceModel Last Checkpoint: 05/10/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
0
25
50
75
100
125
150
175
200
ENG
8:27 PM
5/19/2022

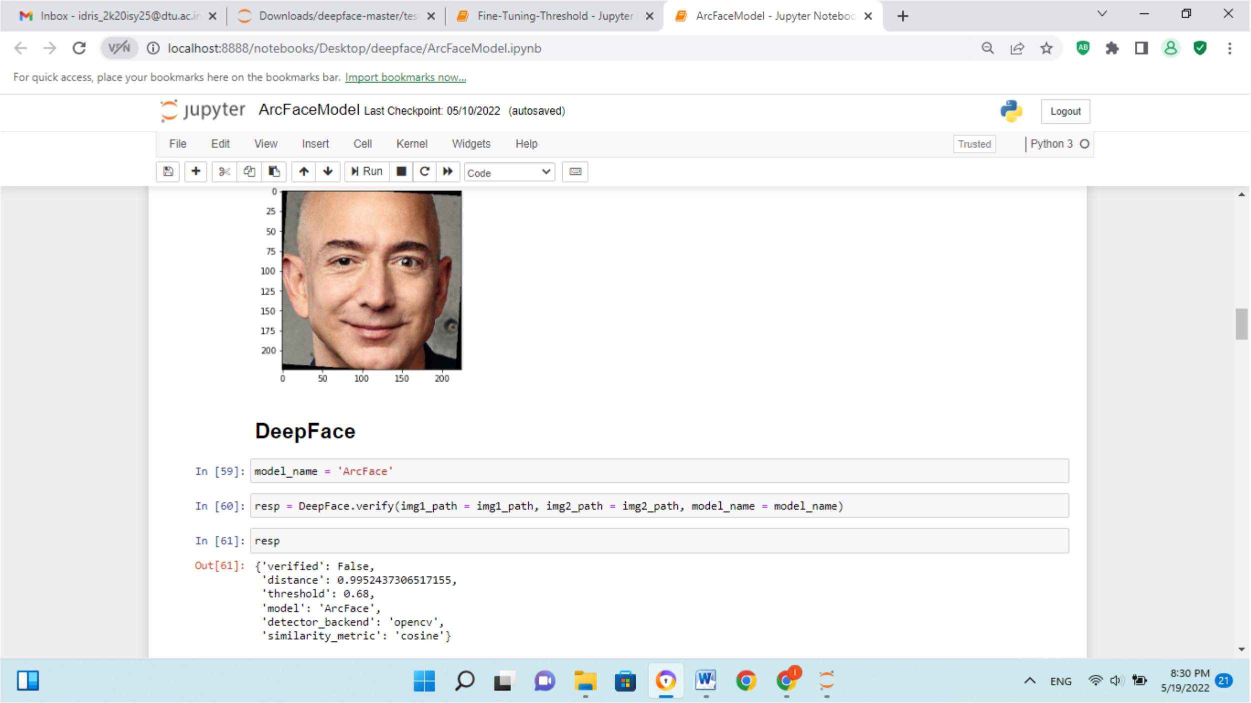
Negative samples

Compare photos of different people

```
In [55]: img1_path = 'tests/dataset/img18.jpg'
         img2_path = 'tests/dataset/img20.jpg'

In [56]: img1 = DeepFace.detectFace(img1_path)
         img2 = DeepFace.detectFace(img2_path)

In [57]: plt.imshow(img1)

Out[57]: <matplotlib.image.AxesImage at 0x19741b171c0>
```

Figure 5.6: Show the images samples in Negative Sample

Here we test our model to find different images of different person (Negative Sample) we use the face detect function to find face in both image.


Inbox - idris_2k20isy25@dtu.ac.in
Downloads/deepface-master/tes
Fine-Tuning-Threshold - Jupyter
ArcFaceModel - Jupyter Notebook
localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
jupyter ArcFaceModel Last Checkpoint: 05/10/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
0
25
50
75
100
125
150
175
200
0
50
100
150
200
ENG
8:30 PM
5/19/2022

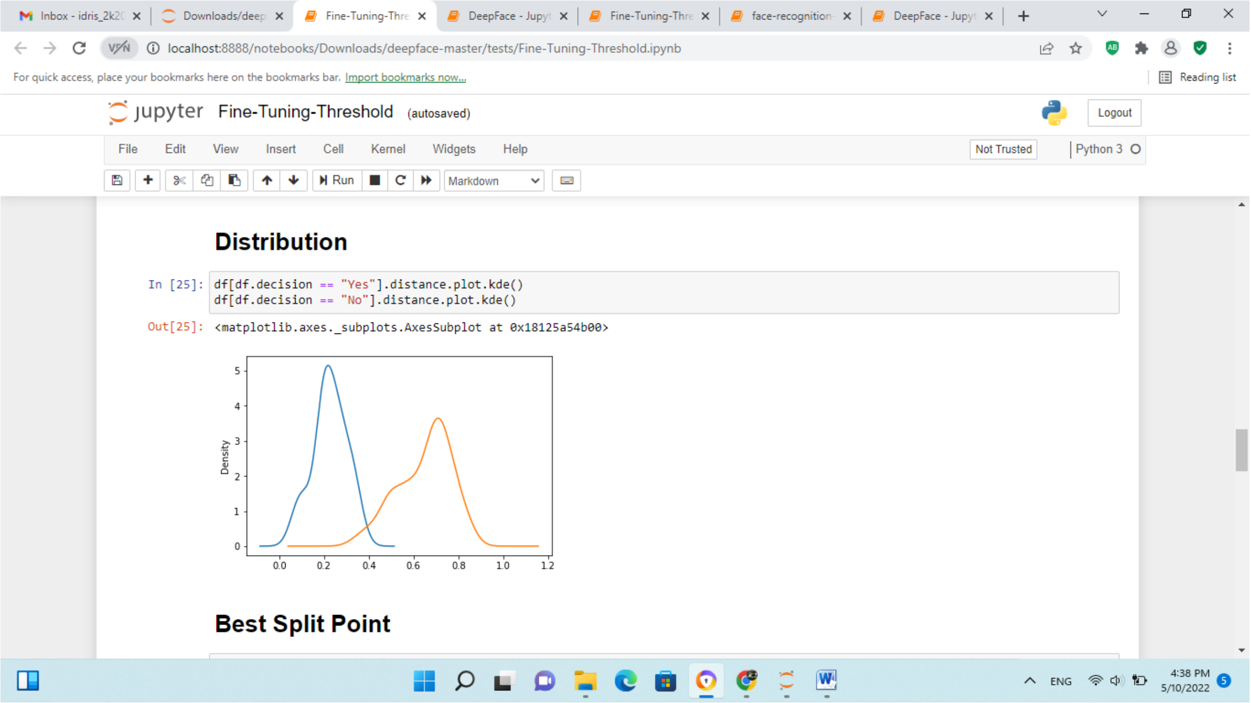
DeepFace

```
In [59]: model_name = 'ArcFace'

In [60]: resp = DeepFace.verify(img1_path = img1_path, img2_path = img2_path, model_name = model_name)

In [61]: resp

Out[61]: {'verified': False,
          'distance': 0.9952437306517155,
          'threshold': 0.68,
          'model': 'ArcFace',
          'detector_backend': 'opencv',
          'similarity_metric': 'cosine'}
```

Figure 5.7: verify different images in Negative Sample

The result of verified samples image is False (Different person).

5.3 ArcFace Model Results


localhost:8888/notebooks/Downloads/deepface-master/tests/Fine-Tuning-Threshold.ipynb
jupyter Fine-Tuning-Threshold (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Density
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
1.2
4:38 PM
5/10/2022

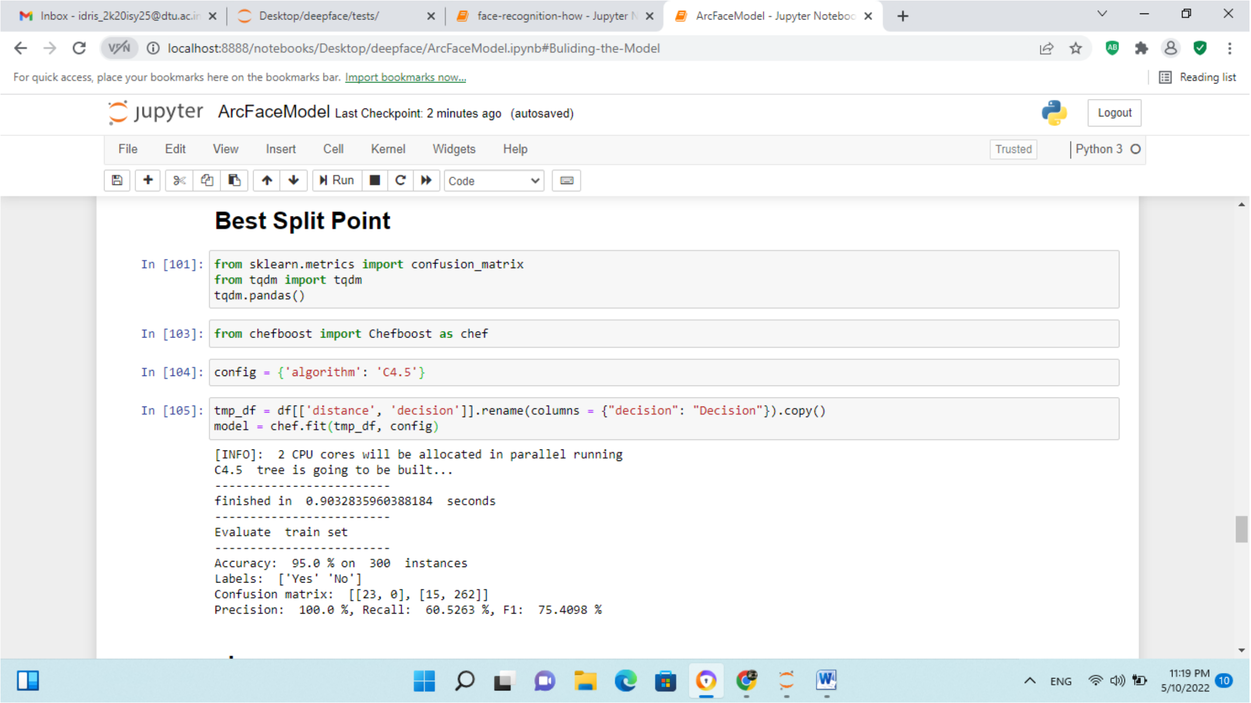
Distribution

```
In [25]: df[df.decision == "Yes"].distance.plot.kde()
         df[df.decision == "No"].distance.plot.kde()

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x18125a54b00>
```

Best Split Point

Figure 5.8: ArcFace Model Distribution

In distribution it seems that (Yes and No) classes are distributed discretely. That's good. Means target labels are unbalanced.

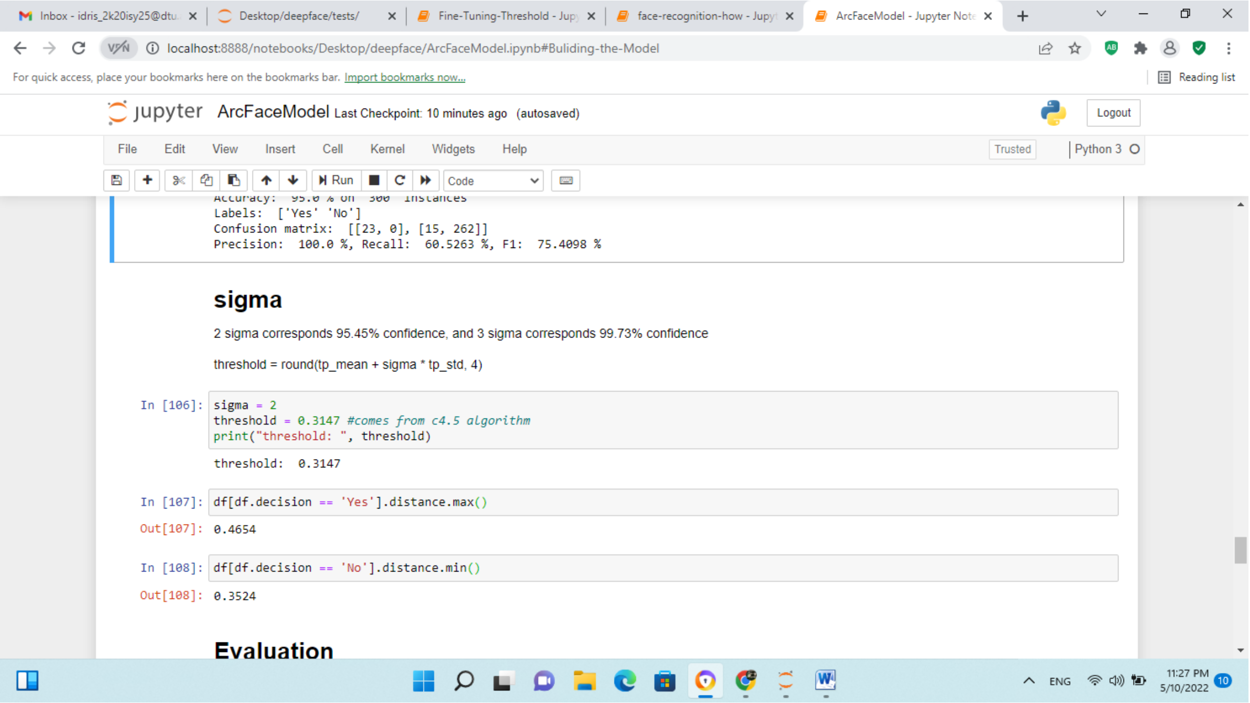

localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb#Buliding-the-Model
jupyter ArcFaceModel Last Checkpoint: 2 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
11:19 PM
5/10/2022

Best Split Point

```
In [101]: from sklearn.metrics import confusion_matrix
          from tqdm import tqdm
          tqdm.pandas()

In [103]: from chefboost import Chefboost as chef

In [104]: config = {'algorithm': 'C4.5'}

In [105]: tmp_df = df[['distance', 'decision']].rename(columns = {"decision": "Decision"}).copy()
          model = chef.fit(tmp_df, config)

          [INFO]:  2 CPU cores will be allocated in parallel running
          C4.5  tree is going to be built...
          -------------------------
          finished in  0.9032835960388184  seconds
          -------------------------
          Evaluate  train set
          -------------------------
          Accuracy:  95.0 % on  300  instances
          Labels:  ['Yes' 'No']
          Confusion matrix:  [[23, 0], [15, 262]]
          Precision:  100.0 %, Recall:  60.5263 %, F1:  75.4098 %
```

Figure 5.9: ArcFace Best Split Point

In the Best Split Point with Accuracy (95.0%) on 300 instances, time for finishing depends on PC Capabilities (Using 2 CPU Cores).



```
Accuracy:  95.0 % on  300  instances
Labels:  ['Yes' 'No']
Confusion matrix:  [[23, 0], [15, 262]]
Precision:  100.0 %, Recall:  60.5263 %, F1:  75.4098 %
```

sigma

2 sigma corresponds 95.45% confidence, and 3 sigma corresponds 99.73% confidence

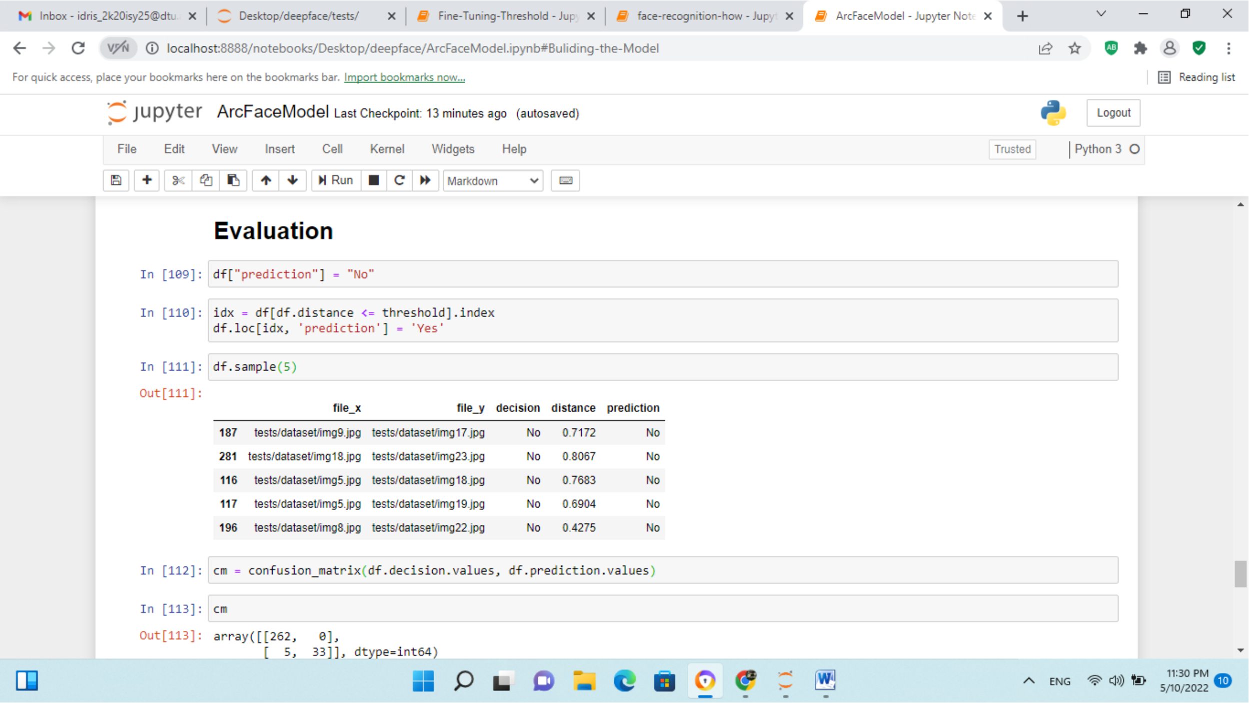
threshold = round(tp_mean + sigma * tp_std, 4)

```
In [106]: sigma = 2
          threshold = 0.3147 #comes from c4.5 algorithm
          print("threshold: ", threshold)

          threshold:  0.3147

In [107]: df[df.decision == 'Yes'].distance.max()

Out[107]: 0.4654

In [108]: df[df.decision == 'No'].distance.min()

Out[108]: 0.3524
```

Evaluation

Figure 5.10: Results of using sigma 2 and 3

This Figure Shows 2 sigma corresponds: 95.45% confidence and 3 sigma correspond: 99.73% confidence.



Evaluation

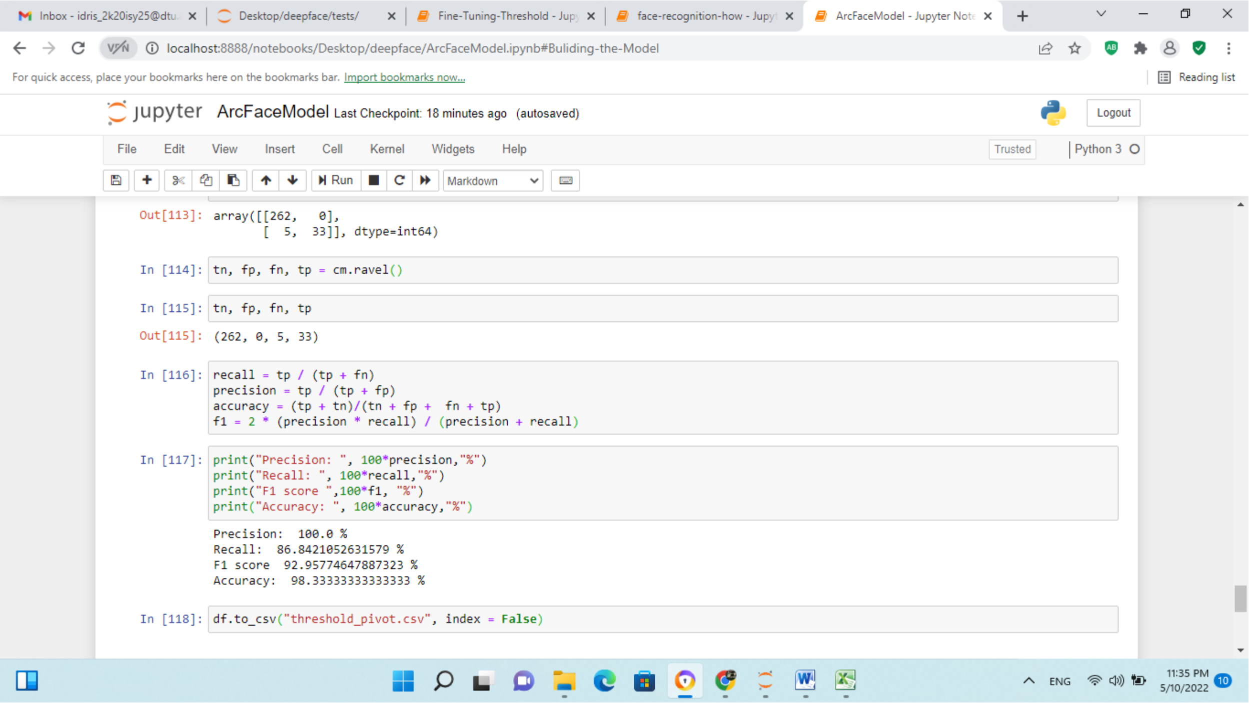
```
In [109]: df["prediction"] = "No"

In [110]: idx = df[df.distance <= threshold].index
          df.loc[idx, 'prediction'] = 'Yes'

In [111]: df.sample(5)

Out[111]:
```

	file_x	file_y	decision	distance	prediction
187	tests/dataset/img9.jpg	tests/dataset/img17.jpg	No	0.7172	No
281	tests/dataset/img18.jpg	tests/dataset/img23.jpg	No	0.8067	No
116	tests/dataset/img5.jpg	tests/dataset/img18.jpg	No	0.7683	No
117	tests/dataset/img5.jpg	tests/dataset/img19.jpg	No	0.6904	No
196	tests/dataset/img8.jpg	tests/dataset/img22.jpg	No	0.4275	No

```
In [112]: cm = confusion_matrix(df.decision.values, df.prediction.values)

In [113]: cm

Out[113]: array([[262,   0],
                 [  5,  33]], dtype=int64)
```

Figure 5.11: Evaluation of images samples

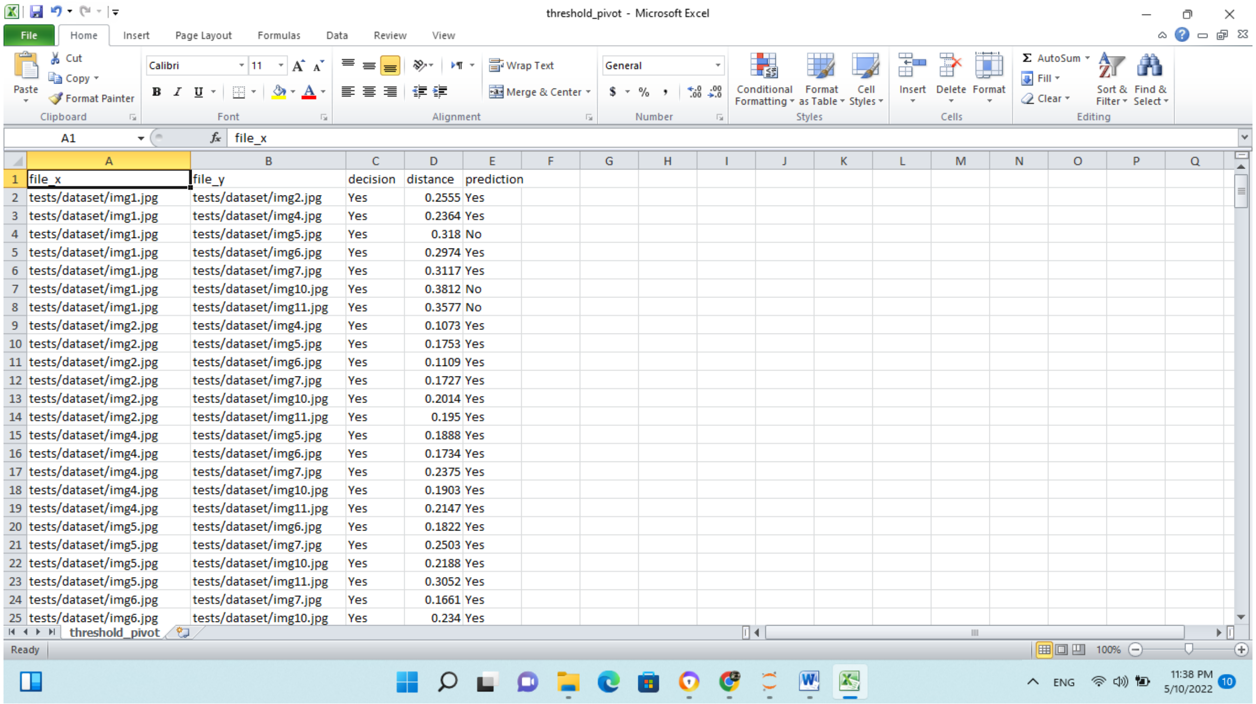
We evaluate five samples images here to see distances, prediction and decision.



```
Out[113]: array([[262,   0],
                 [  5,  33]], dtype=int64)

In [114]: tn, fp, fn, tp = cm.ravel()

In [115]: tn, fp, fn, tp

Out[115]: (262, 0, 5, 33)

In [116]: recall = tp / (tp + fn)
          precision = tp / (tp + fp)
          accuracy = (tp + tn)/(tn + fp +  fn + tp)
          f1 = 2 * (precision * recall) / (precision + recall)

In [117]: print("Precision: ", 100*precision,"%")
          print("Recall: ", 100*recall,"%")
          print("F1 score ",100*f1, "%")
          print("Accuracy: ", 100*accuracy,"%")

          Precision:  100.0 %
          Recall:  86.8421052631579 %
          F1 score  92.95774647887323 %
          Accuracy:  98.33333333333333 %

In [118]: df.to_csv("threshold_pivot.csv", index = False)
```

Figure 5.12: Evaluation of all images samples in dataset

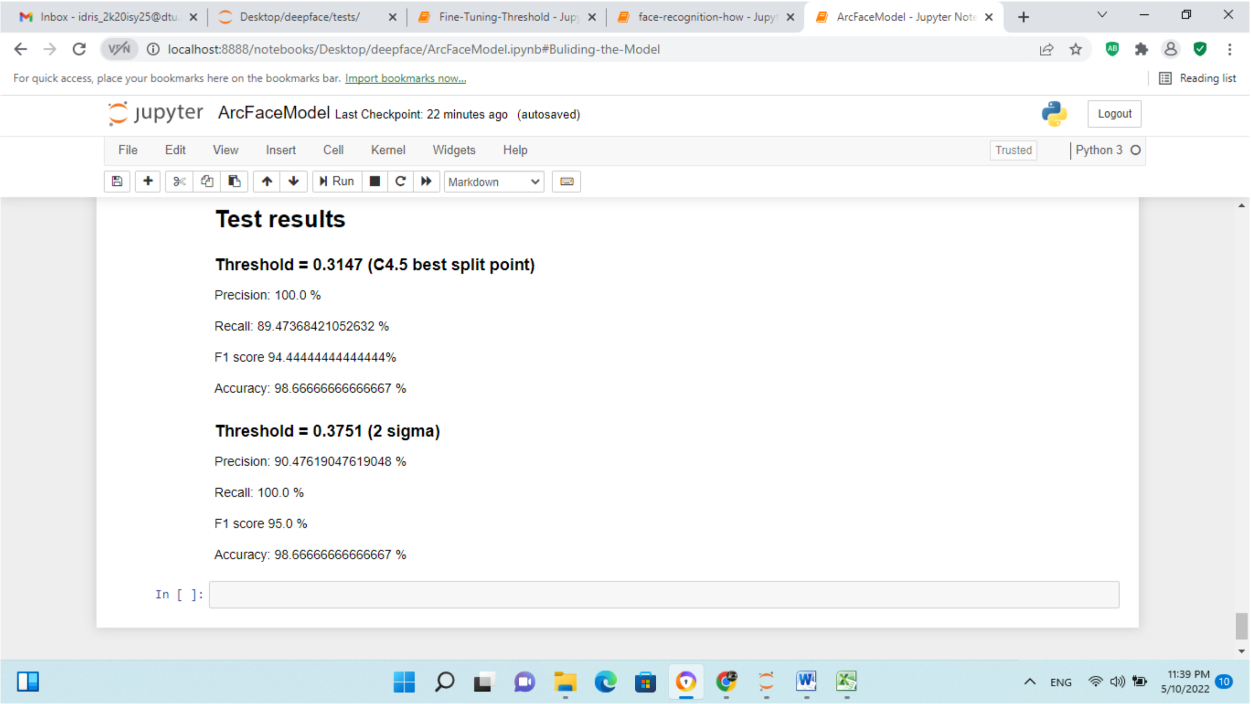
We evaluate all samples images in our dataset and exporting in excel sheet (threshold_pivot.csv) see distances, prediction and decision.



threshold_pivot - Microsoft Excel

	A	B	C	D	E
1	file_x	file_y	decision	distance	prediction
2	tests/dataset/img1.jpg	tests/dataset/img2.jpg	Yes	0.2555	Yes
3	tests/dataset/img1.jpg	tests/dataset/img4.jpg	Yes	0.2364	Yes
4	tests/dataset/img1.jpg	tests/dataset/img5.jpg	Yes	0.318	No
5	tests/dataset/img1.jpg	tests/dataset/img6.jpg	Yes	0.2974	Yes
6	tests/dataset/img1.jpg	tests/dataset/img7.jpg	Yes	0.3117	Yes
7	tests/dataset/img1.jpg	tests/dataset/img10.jpg	Yes	0.3812	No
8	tests/dataset/img1.jpg	tests/dataset/img11.jpg	Yes	0.3577	No
9	tests/dataset/img2.jpg	tests/dataset/img4.jpg	Yes	0.1073	Yes
10	tests/dataset/img2.jpg	tests/dataset/img5.jpg	Yes	0.1753	Yes
11	tests/dataset/img2.jpg	tests/dataset/img6.jpg	Yes	0.1109	Yes
12	tests/dataset/img2.jpg	tests/dataset/img7.jpg	Yes	0.1727	Yes
13	tests/dataset/img2.jpg	tests/dataset/img10.jpg	Yes	0.2014	Yes
14	tests/dataset/img2.jpg	tests/dataset/img11.jpg	Yes	0.195	Yes
15	tests/dataset/img4.jpg	tests/dataset/img5.jpg	Yes	0.1888	Yes
16	tests/dataset/img4.jpg	tests/dataset/img6.jpg	Yes	0.1734	Yes
17	tests/dataset/img4.jpg	tests/dataset/img7.jpg	Yes	0.2375	Yes
18	tests/dataset/img4.jpg	tests/dataset/img10.jpg	Yes	0.1903	Yes
19	tests/dataset/img4.jpg	tests/dataset/img11.jpg	Yes	0.2147	Yes
20	tests/dataset/img5.jpg	tests/dataset/img6.jpg	Yes	0.1822	Yes
21	tests/dataset/img5.jpg	tests/dataset/img7.jpg	Yes	0.2503	Yes
22	tests/dataset/img5.jpg	tests/dataset/img10.jpg	Yes	0.2188	Yes
23	tests/dataset/img5.jpg	tests/dataset/img11.jpg	Yes	0.3052	Yes
24	tests/dataset/img6.jpg	tests/dataset/img7.jpg	Yes	0.1661	Yes
25	tests/dataset/img6.jpg	tests/dataset/img10.jpg	Yes	0.234	Yes

Figure 5.13: Evaluation of all images samples in Excel Sheet Format


Inbox - idris_2k20isy25@dtu
Desktop/deepface/tests/
Fine-Tuning-Threshold - Jupy
face-recognition-how - Jupy
ArcFaceModel - Jupyter Note
localhost:8888/notebooks/Desktop/deepface/ArcFaceModel.ipynb#Buliding-the-Model
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Reading list
jupyter
ArcFaceModel Last Checkpoint: 22 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Markdown
In [ ]:
ENG
11:39 PM
5/10/2022

Test results

Threshold = 0.3147 (C4.5 best split point)

Precision: 100.0 %

Recall: 89.47368421052632 %

F1 score 94.44444444444444%

Accuracy: 98.66666666666667 %

Threshold = 0.3751 (2 sigma)

Precision: 90.47619047619048 %

Recall: 100.0 %

F1 score 95.0 %

Accuracy: 98.66666666666667 %

Figure 5.14: The Test Result of the Model

We see the comparison of results between Threshold of best split point and threshold of 2 sigma we found that Threshold = 0.3147 (C4.5 best split point) and Threshold = 0.3751 (2 sigma) with accuracy 98.6% for both.

6. CONCLUSION

With The increasing use of technology represented facial recognition system now during video surveillance by security authorities, or at electronic gates at Security sensitive organizations. From one side, it brings immense advantage to the business and end-users to helps them to enhance their security and track down the trespassers. And from the opposite side, it should be misused for private benefit and lead to some serious consequences. It's good that there's another a part of the system called a facial recognition service, which allows comparison between the photo of an unknown person and therefore the various government records, to help within the search for their true identity.

This Thesis presented a brief survey of issues methods and applications in area of face recognition. There's much work to be done in order to understand methods that reflect how humans recognize faces and optimally make use of the temporal evolution of the appearance of the face for recognition. We hope that this project will further encourage during this field to participate and pay more attention to the utilization of local techniques for face recognition technology. Finally, Technology for sure has matured and is slowly settling into the lives of people due to its usability and has made lives easier. Right from its invention to its accessibility, artificial intelligence is here not just to remain, but to grow and conquer.

Areas for Future Work

Facial recognition technologies have become the modern day city and one of the most popular technologies that will change our daily life. Face recognition technology is expected to grow and flourish further, and generate huge revenues in the coming years.

- It is expected that this new system is designed with strong safeguards to protect privacy. It is clear that institutions will obtain permission to enter this system through partnership agreements, and there will be clear restrictions on the ability of the private sector to access the data within this system.
- One of the requirements is to provide guarantees to legalize private institutions' access to private citizens' data. By identifying appropriate types of conditions for implementing privacy enhancing technologies. Identifying privacy and bias considerations in the government acquisition and explore the targets of this technology's privacy expectations.
- It is also possible to implement part of this advanced system, which is the face verification service. This service allows verification of the face of each individual based on his image, linking the image of the person with his other image in one of the government records.

7. REFERENCES

1. Er M. J., Wu S., Lu J. and Toh H. L., “Face recognition with radial basis function (RBF) neural networks”, IEEE Trans. Neural Networks, vol. 13, no. 3, pp. 697-710
2. Zhao, W., Chellappa, R., Phillips, P.J. and Rosenfeld, A., 2003. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, *35*(4), pp.399-458.
3. Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, present, and future of face recognition: A review. Electronics, 9(8), 1188.
4. P. Nagesh and B. Li, “*A compressive sensing approach for expression invariant face recognition*” in IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1518–1525
5. S. Liao, A. K. Jain, and S. Z. Li, “*Partial face recognition: Alignment free approach*” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 5, pp. 1193–1205, 2013.
6. Q. Yin, X. Tang, and J. Sun, “*An associate-predict model for face recognition*” in IEEE Conference on Computer Vision and Pattern Recognition, June 2011, pp. 497–504.
7. N. M. Ara, N. S. Simul and M. S. Islam, "Convolutional neural network approach for vision based student recognition system," *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1-6, doi: 10.1109/ICCITECHN.2017.8281789
8. Kiela, Douwe, and Léon Bottou. "Learning image embeddings using convolutional neural networks for improved multi-modal semantics." *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*. 2014.
9. T. Baltrusaitis, A. Zadeh, Y. C. Lim and L. Morency, "OpenFace 2.0: Facial Behavior Analysis Toolkit," *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018, pp. 59-66, doi: 10.1109/FG.2018.00019
10. Kortli Y, Jridi M, Al Falou A, Atri M. Face recognition systems: A survey. Sensors. 2020 Jan;20(2):342.
11. Mustafa, Ahmed Shamil. "Face Recognition Systems Using Different Algorithms: A Literature." Australian Journal of Basic and Applied Sciences 11.7 (2017): 9-17.

12. G. Jetsiktat, S. Panthuwadeethorn and S. Phimoltares, "Enhancing user authentication of online credit card payment using face image comparison with MPEG7-edge histogram descriptor," 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2015, pp. 67-74, doi: 10.1109/ICIIBMS.2015.7439481

13. J. R. D. Kho and L. A. Vea, "Credit card fraud detection based on transaction behavior," TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 1880-884, doi: 10.1109/TENCON.2017.8228165.

14. Liu, Yu-li, Wenjia Yan, and Bo Hu. "Resistance to facial recognition payment in China: The influence of privacy-related factors." Telecommunications Policy 45.5 (2021): 102155.

15. ZHANG, Lin-Lin, et al. A Study on the Impact of Face Recognition Payment System Characteristics and Innovation Resistance on Intention to Use: Focusing on Chinese Users. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021, 12.10: 1005-1013.

16. Al Farawn, Ali, et al. "Secured e-payment system based on automated authentication data and iterated salted hash algorithm." Telkomnika 18.1 (2020): 538-544.

17. Mehta, Vishakha, and Mayank Patel. "Implementing Banking and Payment System using Face Detection and Recognition Method." International Journal of Innovative Science and Research Technology ISSN No:-2456-2165

18. Deng, J., Guo, J., Xue, N. and Zafeiriou, S., 2019. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4690-4699).

19. Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. "Face recognition with local binary patterns." European conference on computer vision. Springer, Berlin, Heidelberg, 2004.

20. Deng, Jiankang, and Stefanos Zafeririou. "Arcface for disguised face recognition." Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019.

PAPER NAME

Thesis_2K20_ISY_25.pdf

WORD COUNT	CHARACTER COUNT
7595 Words	44714 Characters
PAGE COUNT	FILE SIZE
40 Pages	3.7MB
SUBMISSION DATE	REPORT DATE
May 19, 2022 10:12 PM GMT+5:30	May 19, 2022 10:14 PM GMT+5:30

● 17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 10% Internet database
- Crossref database
- 15% Submitted Works database
- 6% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 10 words)