REPRESENTATION LEARNING

Project Report

On

FACE RECOGNITION USING PCA

Submitted For **3rd Semester**

Department of Data Science And Artificial Intelligence

By

Harshita Panchbhai (201020417)

Ankit Singh (201020407)

Pallavi Baghel (201020438)

Under the guidance of **Dr. ABHISHEK SHARMA**

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology, Naya Raipur

ACKNOWLEDGEMENT

At the outset, we would like to express our wholehearted and deep sense of gratitude to our guide Dr.Abhishek Sharma for his guidance, help and encouragement throughout our research work. We greatly admire his attitude towards research , creative thinking , hard work and dedication in work. We are highly grateful to him for patiently checking all our models. This project would not have been possible without his bounteous efforts. More than a guide ,he is our mentor for shaping our personal and professional life without whom we would not have been where we are today. We owe our profound gratitude to Dr. Abhishek Sharma for his support in all respects.

Harshita Panchbhai (201020417)

Ankit Singh (201020407)

Pallavi Baghel (201020438)

ABSTRACT

Face detection is the process of identifying a human face whereas face recognition is the process of verification of any known person's face from the system's database.Thus human face detection and recognition is the process of identification and verification of a known person's face by providing input of that person's facial image. This technique makes it possible to use the facial images of a person to authenticate him into a secure system, for criminal identification, for passport verification, terrorist detection and so on. Human face is a complex multidimensional structure and needs good computing techniques for the recognition.Our approach treats face recognition as a two dimensional recognition problem. In this research paper face recognition is done by Principal Component Analysis (PCA) algorithm. This is a personal identification system that uses personal characteristics of a person's face to identify that person's face.In this paper the implementation is also done.

INTRODUCTION

Face is a complex multidimensional structure and needs good computing techniques for recognition. Our approach treats face recognition as a two-dimensional recognition problem. In this scheme face recognition is done by Principal Component Analysis (PCA). Face images are projected onto a face space that encodes best variation among known face images. The face space is defined by eigenfaces which are eigenvectors of the set of faces, which may not correspond to general facial features such as eyes, nose, lips. The Eigenface approach uses the PCA for recognition of the images. The system performs by projecting pre extracted face images onto a set of face space that represent significant variations among known face images. Face will be categorized as known or unknown face after matching with the present database. If the user is new to the face recognition system then his/her template will be stored in the database and matched against the templates stored in the database. The variable reducing theory of PCA accounts for the smaller face space than the training set of faces.

FACE RECOGNITION PROCESS

One of the simplest and most effective PCA approaches used in face recognition systems is the so-called eigenface approach. This approach transforms faces into a small set of essential characteristics, eigenfaces, which are the main components of the initial set of learning images (training set). Recognition is done by projecting a new image in the eigenface subspace, after which the person is classified by comparing its position in eigenface space with the position of known individuals. The advantage of this approach over other face recognition systems is in its simplicity, speed and insensitivity to small or gradual changes on the face. The problem is limited to files that can be used to recognize the face. Namely, the images must be vertical frontal views of human faces. The whole recognition process involves two steps:

A. Initialization process

B. Recognition process

The Initialization process involves the following operations:

i. Acquire the initial set of face images called as training set.

ii. Calculate the Eigenfaces from the training set, keeping only the highest eigenvalues. These M images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.

iii. Calculate distribution in this M-dimensional space for each known person by projecting his or her face images onto this face-space.

These operations can be performed from time to time whenever there is a free excess operational capacity. This data can be cached which can be used in the further steps eliminating the overhead of re-initializing, decreasing execution time thereby increasing the performance of the entire system.

Having initialized the system, the next process involves the steps:

i. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the Eigenfaces.

ii. Determine if the image is a face at all (known or unknown) by checking to see if the image is sufficiently close to a ―free space.

iii. If it is a face, then classify the weight pattern as either a known person or as unknown.

iv. Update the eigenfaces or weights as either a known or unknown, if the same unknown person face is seen several times then calculate the characteristic weight pattern and incorporate into known faces.

The last step is not usually a requirement of every system and hence the steps are left optional and can be implemented as when there is a requirement.

LITERATURE SURVEY

Principal Component Analysis (PCA)

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. PCA is a variable reduction procedure and useful when obtained data have some redundancy. This will result in reduction of variables into smaller numbers of variables which are called Principal Components which will account for the most of the variance in the observed variable. Problems arise when we wish to perform recognition in a high-dimensional space. Goal of PCA is to reduce the dimensionality of the data by retaining as much variation as possible in our original data set. On the other hand dimensionality reduction implies information loss. The best low-dimensional space can be determined by the best principal components.

The major advantage of PCA is using it in an eigenface approach which helps in reducing the size of the database for recognition of test images. The images are stored as their feature vectors in the database which are found out projecting each and every trained image to the set of Eigen faces obtained. PCA is applied on the Eigenface approach to reduce the dimensionality of a large data set.

EigenFace Approach

It is an adequate and efficient method to be used in face recognition due to its simplicity, speed and learning capability. Eigenfaces are a set of Eigenvectors used in the Computer Vision problem of human face recognition. They refer to an appearance based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner. The Eigen faces are Principal Components of a distribution of faces, or equivalently, the Eigenvectors of the covariance matrix of the set of the face images, where an image with N by N pixels is considered a point in N 2 dimensional space. Previous work on face recognition ignored the issue of face stimulus, assuming that predefined measurements were relevant and sufficient. This suggests that coding and decoding of face images may give information of face images emphasizing the significance of features. These features may or may not be related to facial features such as eyes, nose, lips and hairs. We want to extract the relevant information in a face image, encode it efficiently and compare one face encoding with a database of faces encoded similarly. A simple approach to extracting the information content in an image of a face is to somehow capture the variation in a collection of face images. We wish to find Principal Components of the distribution of faces, or the Eigenvectors of the covariance matrix of the set of face images. Each image location contributes to each Eigenvector, so that we can display the Eigenvector as a sort of face. Each face image can be represented exactly in terms of linear combination of the Eigen faces. The number of possible Eigenfaces is equal to the number of face images in the training set. The faces can also be approximated by using best Eigen faces, those that have the largest Eigenvalues, and which therefore account for most variance between the set of face images. The primary reason for using fewer Eigenfaces is computational efficiency.

Eigenvalues and EigenVectors

In linear algebra, the eigenvectors of a linear operator are non-zero vectors which, when operated by the operator, result in a scalar multiple of them.

Scalar is then called Eigenvalue (λ) associated with the eigenvector (X).

Eigenvector is a vector that is scaled by linear transformation.

It is a property of the matrix. When a matrix acts on it, only the vector magnitude is changed, not the direction.

$AX = \lambda X$, where A is a vector function.

$(A - \lambda I)X = 0$, where I is the identity matrix.

This is a homogeneous system of equations and forms fundamental linear algebra. We know a non-trivial solution exists if and only if

$Det(A - \lambda I) = 0$, where det denotes determinant.

When evaluated, it becomes a polynomial of degree n. This is called the characteristic polynomial of A.

If A is N by N then there are n solutions or n roots of the characteristic polynomial.

Thus there are n Eigenvalues of A satisfying the equation.

$AXi = \lambda iXi$, where i = 1,2,3,.....n

If the Eigenvalues are all distinct, there are n associated linearly independent eigenvectors, whose directions are unique, which span an n dimensional Euclidean space.

IMPLEMENTATION

I. Testing Parameters

A colored face image is converted to grayscale image as grayscale images are easier for applying computational techniques in image processing.

II. Image Representation

1. **Convert images of training set to image vectors**

Let us consider that the training set contains **M** images. Each image is of size **NxN**. Rows of the pixels in an **(NxN)** image are placed one after the other to form a one dimensional vector of dimensions **(N^2x1)**.

2. **Normalize the face vectors.**

   2.1 Calculate the average face vectors.

   2.2 Subtract the average face vector from each face vector.

3. **Calculate the Eigenvectors (Eigenvectors represent variations in faces)**

   However, we need to reduce the dimensionality first. So, the solution to this problem is in the next step.

4. **Calculate eigenvectors from reduced covariance matrix.**
5. **Select K best eigenfaces such that K<=M and can represent the whole training set.**
6. **Convert lower dimension K eigenvectors to original face dimensionality.**
7. **Represent each face as a linear combination of all K eigenvectors.**
8. **Now test on an unknown face and obtain results.**

APPLICATIONS

Biometrics

Biometrics is used in the process of authentication of a person by verifying or identifying that a user requesting a network resource is who he, she, or it claims to be, and vice versa. It uses the property of a human trait associated with a person itself like structure of the finger, face details etc. By comparing the existing data with the incoming data we can verify the identity of a particular person . There are many types of biometric systems like fingerprint recognition, face detection and recognition, iris recognition etc.. These traits are used for human identification in surveillance system, criminal identification. Advantages of using these traits for identification are that they cannot be forgotten or lost. These are unique features of a human being which is being used widely

Automobile security

Although you probably don't spend too much time thinking about them, every now and then you've probably seen an armored truck cruising around town. These trucks often carry important items, whether that's intel or cash, and rely on facial recognition technology to prevent theft or even ensure that the driver's eyes are on the road.

In another capacity, facial recognition technology is sometimes used by ride-sharing apps to confirm that a given passenger is who they say they are. Or alternatively, the same technology can guarantee that the passenger is approaching the right driver.

Access Control

Outside of cars and smartphones, facial recognition can be used in the home to grant access to certain IoT devices in addition to entry into the home itself. As this technology becomes more and more advanced, people will feel better protected against home invasions and robberies.

Immigration

Immigration offices exist as an extension to more well-known government segments. Facial recognition technology is used to enforce stricter border control, particularly when it comes to criminals and persons of interest who attempt to cross the border.

Education

Other than federal and local security, facial recognition applications may exist most prominently in the education sector.

A growing number of schools already use cameras that utilize facial recognition software to identify students, staff, unauthorized individuals, and even behavior that could present a threat to safety. This is one of many new tech trends that are transforming education. For schools using this technology, the main benefit they see is tracking student attendance as well as maintaining the security of their campus. Unfortunately, technology can be very biased and studies have shown evidence for the software to be banned.

Retail

Though the United States is behind in regards to using facial recognition for retail, other countries like Japan have been doing so for quite some time. For example, vending machines in Japan can recommend drinks for the consumer by using facial recognition technology to approximate a customer's gender and age. Lately, it looks as if the USA is catching up. Amazon opened its first Amazon Go store in 2018. There is no check-out and the store depends entirely on sensors to figure out what a customer picks up and buys. Given this, it might be safe to assume that Amazon has the brainpower to start mapping shopping behaviors in the near future once online shoppers are connected to offline faces.

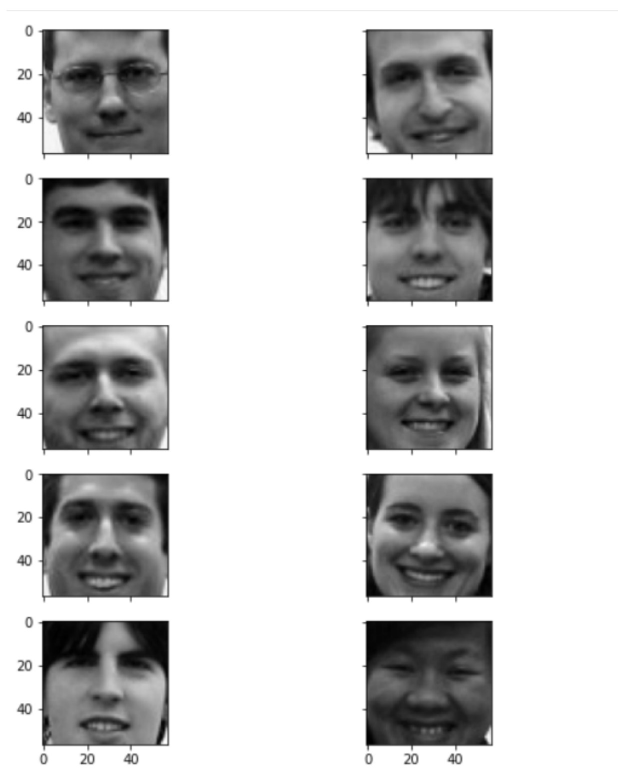
Healthcare

Applications of facial recognition technology are used in hospitals, especially those working in assisted living. The software serves to keep track of everything that is going on within a hospital, ensuring patients are safe and the premise is secure.

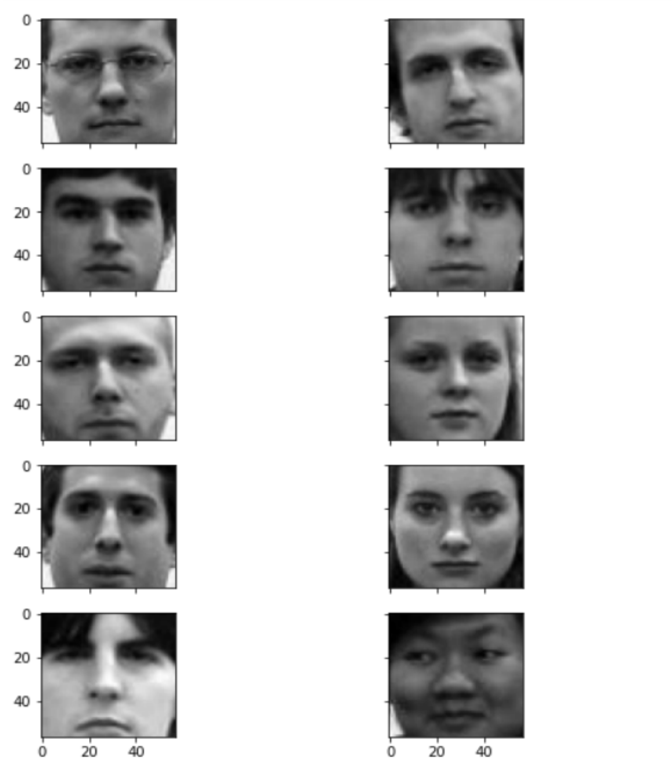
If a patient wanders away from the care-giving facility with no identification, facial recognition can help quickly identify and find them quickly to prevent any harm from coming to them.

OBSERVATIONS FROM OUR DATASET

Training Images(57x57) :



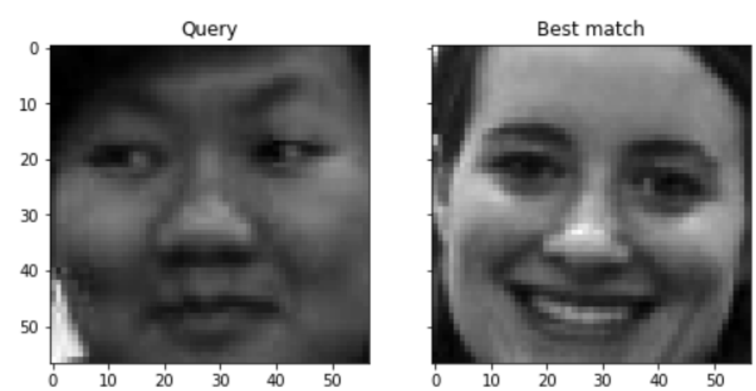
TEST Images(57x57) :

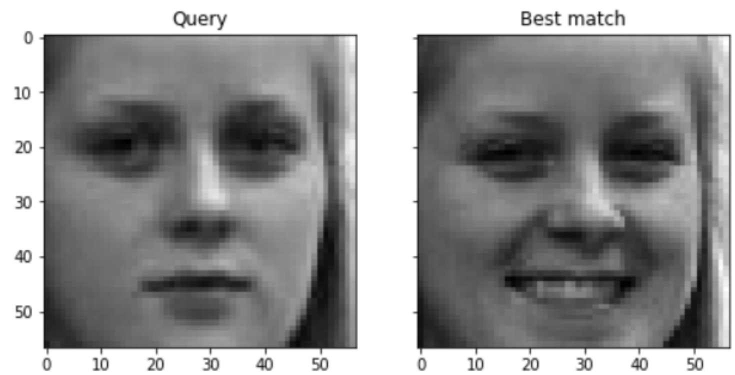


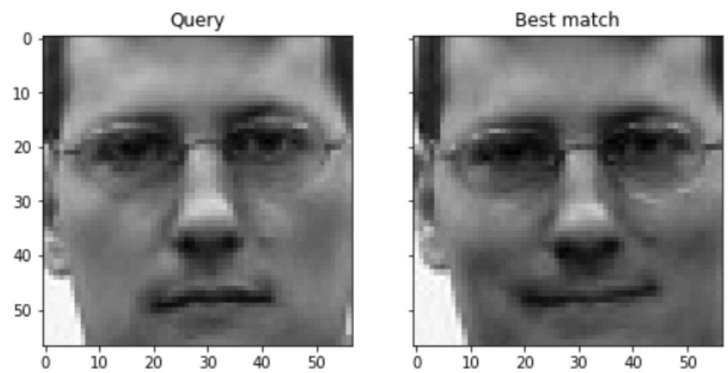
RESULTS

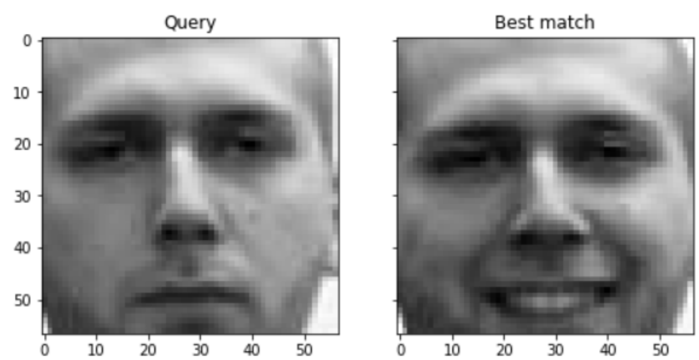
1. **For Principal Components equal to 1,2,3 and 4:**

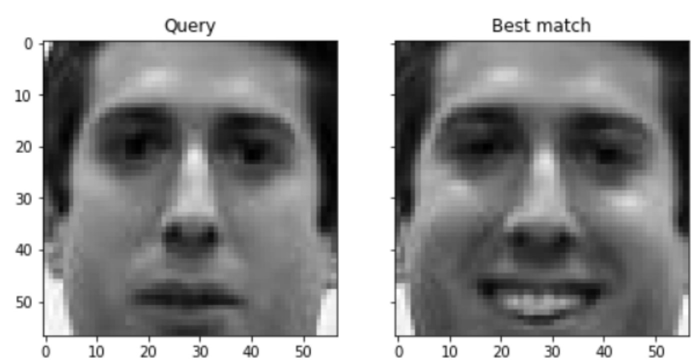
   (Here we have output for Principal Components =3)

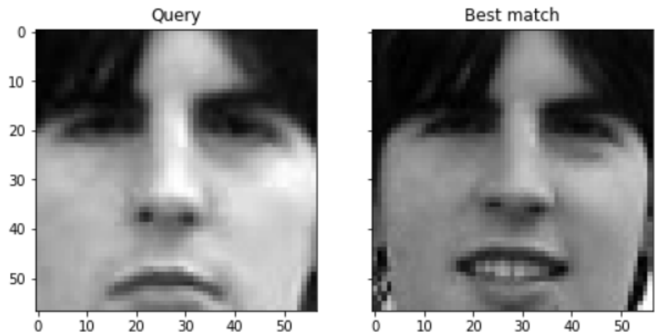


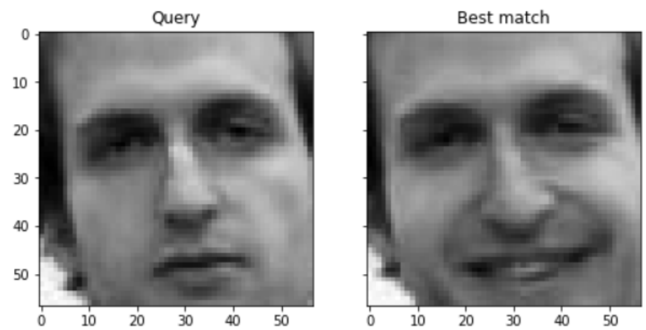

Query
Best match

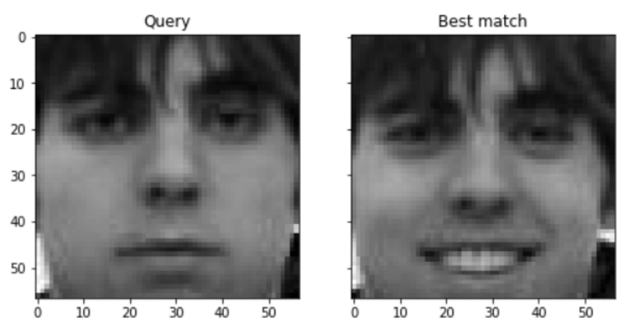

Query
Best match

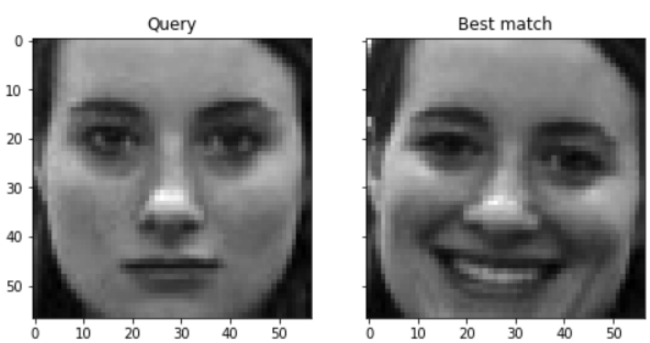

Query
Best match

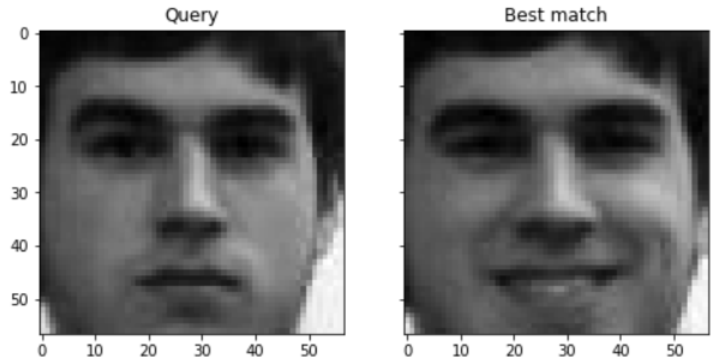

Query
Best match


Query
Best match

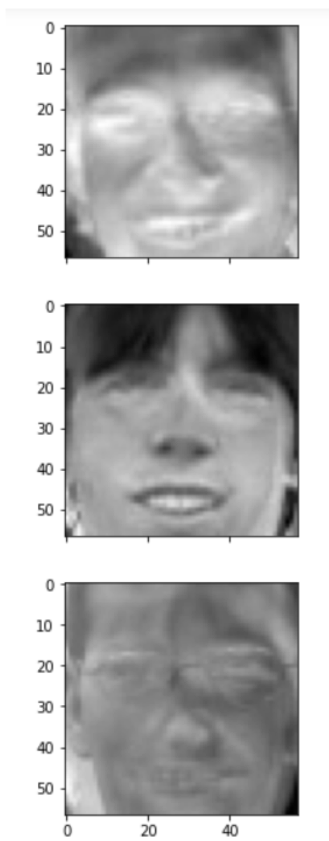

Query
Best match


Query
Best match

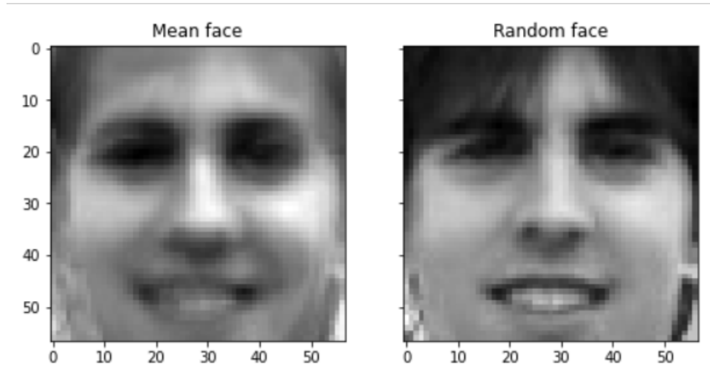

Query
Best match



Eigenfaces Obtained:

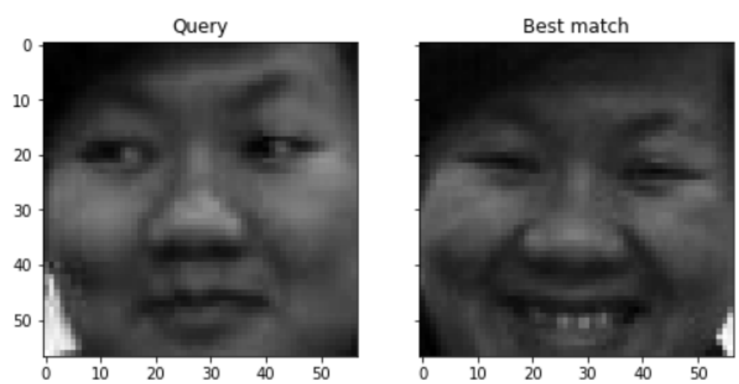


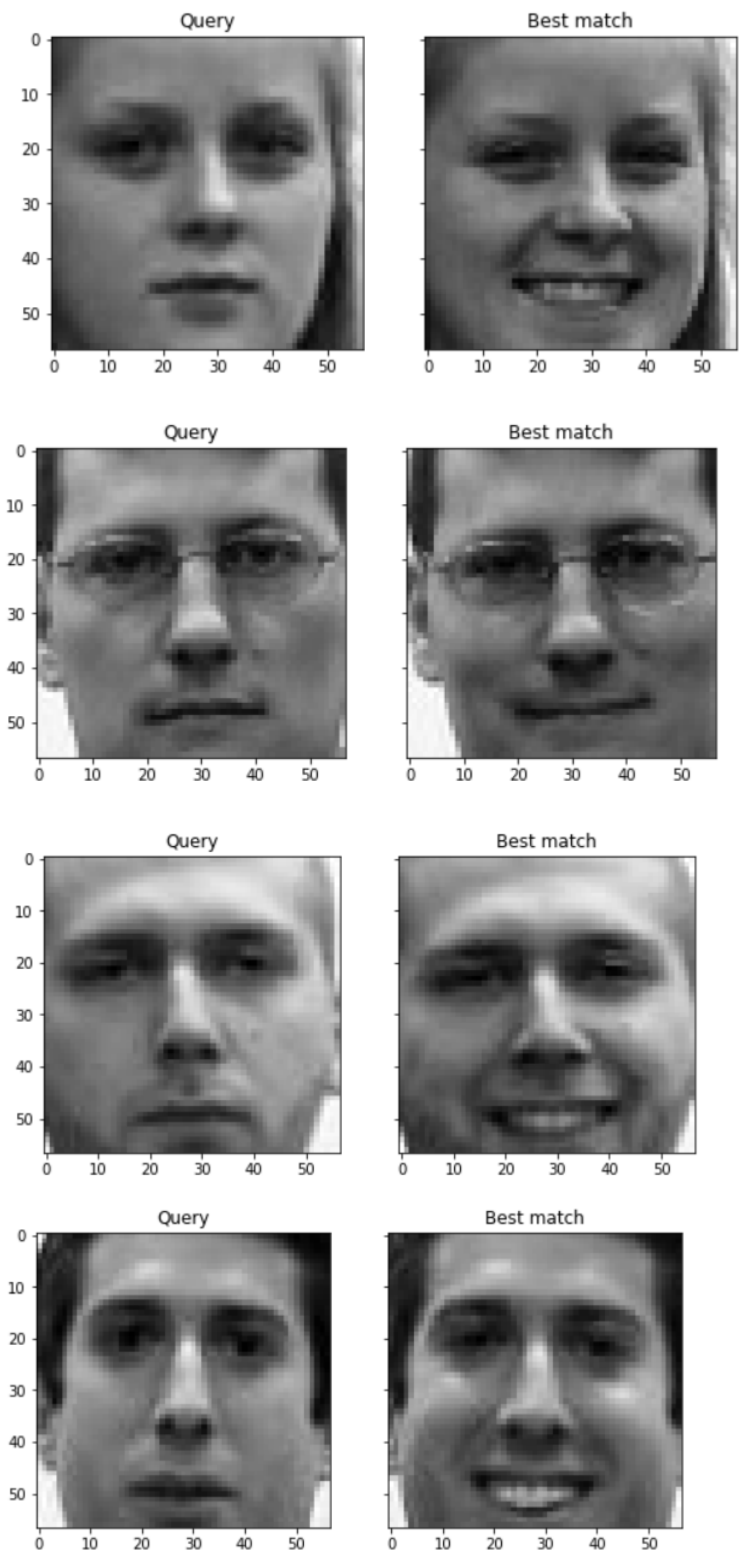
Mean and Random Faces:

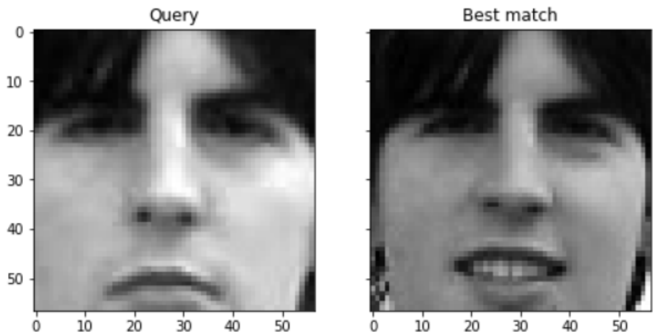


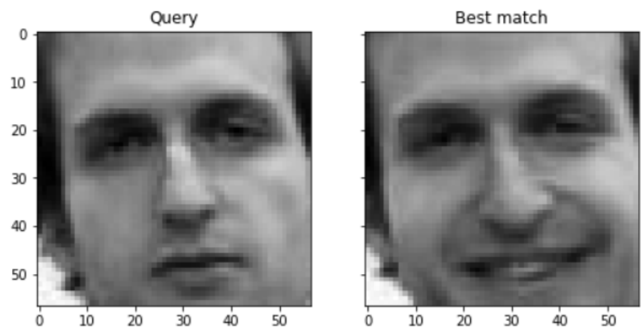
2. For Principal Components equal to 5,6,7,8,9 and 10:

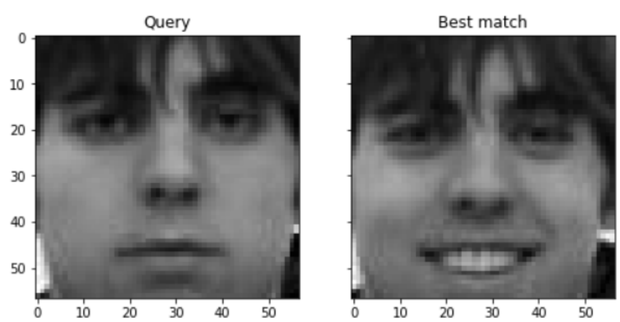
(Here we have output for Principal Components = 5)

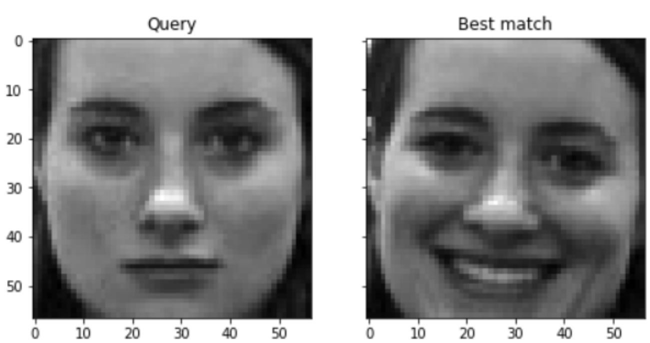


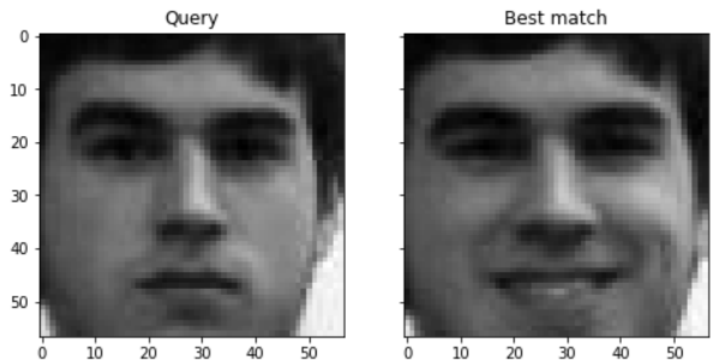

Query
Best match
Query
Best match
Query
Best match
Query
Best match


Query
Best match
0
10
20
30
40
50

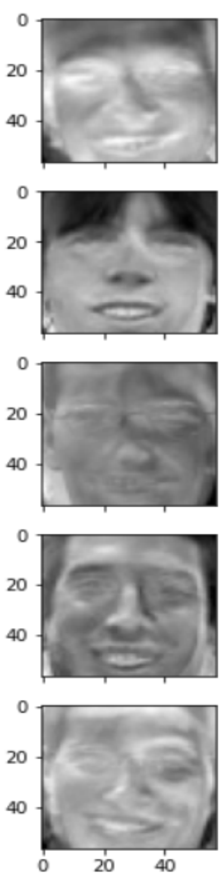

Query
Best match
0
10
20
30
40
50


Query
Best match
0
10
20
30
40
50

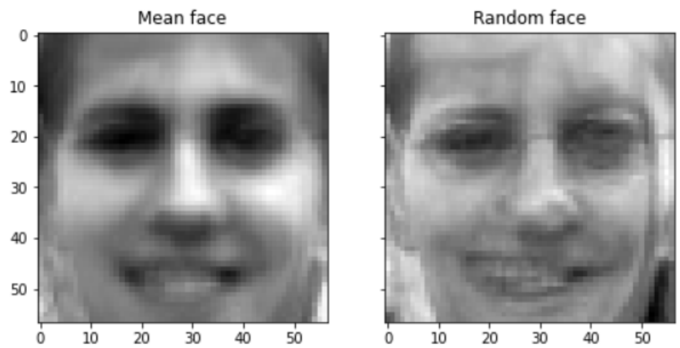

Query
Best match
0
10
20
30
40
50



Eigenfaces Obtained:



Mean and Random Faces:



Hence, by experimenting values of Principal Components from 1 to 10, we find that values below 5 predict with 90% accuracy and values from 5 and above give 100% accurate results.

PC = 5 gives least Euclidean Distances on all faces.

CONCLUSION

Face recognition technology has come a long way in the last twenty years. Today, machines are able to automatically verify identity information for secure transactions, for surveillance and security tasks, and for access control to buildings etc. These applications usually work in controlled environments and recognition algorithms can take advantage of the environmental constraints to obtain high recognition accuracy. However, next generation face recognition systems are going to have widespread application in smart environments -- where computers and machines are more like helpful assistants.To achieve this goal computers must be able to reliably identify nearby people in a manner that fits naturally within the pattern of normal human interactions. They must not require special interactions and must conform to human intuitions about when recognition is likely. This implies that future smart environments should use the same modalities as humans, and have approximately the same limitations. These goals now appear in reach -- however, substantial research remains to be done in making person recognition technology work reliably, in widely varying conditions using information from single or multiple modalities.

CODE

```
# Import matplotlib and seaborn library
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.cm as cm
import seaborn as sns
# Import scikit-learn library
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.datasets import fetch_lfw_people
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.svm import SVC
from skimage import data,io,transform
import zipfile
import cv2
import math
import numpy as np
import pandas as pd
from PIL import Image

faces = {}
# Read the zip file containing the dataset
with zipfile.ZipFile("faces_dataset.zip") as facezip:
    for filename in facezip.namelist():
```

```
        if not filename.endswith(".pgm"):
            x = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
            continue # not a face picture
    with facezip.open(filename) as image:
    # If we extracted files from zip, we can use cv2.imread(filename) instead
        faces[filename] = cv2.imdecode(np.frombuffer(image.read(), np.uint8), cv2.IMREAD_GRAYSCALE)


# Training Data
# Smiling faces of 10 people are last 10 images of the dataset
fig, axes = plt.subplots(5,2,sharex=True,sharey=True,figsize=(8,10))
faceimages = list(faces.values())[-10:] # take last 10 images
for i in range(10):
    axes[i%5][i//5].imshow(faceimages[i], cmap="gray")
plt.show()


# Testing Data
# Normal faces of same 10 people are first 10 images in the dataset
fig, axes = plt.subplots(5,2,sharex=True,sharey=True,figsize=(8,10))
faceimages = list(faces.values())[0:10] # take first 10 images
for i in range(10):
    axes[i%5][i//5].imshow(faceimages[i], cmap="gray")
plt.show()
```

```
faceshape = list(faces.values())[0].shape
print("Face image shape:", faceshape)

print(list(faces.keys())[:21])

classes = set(filename.split("/")[0] for filename in faces.keys())
print("Number of classes:", len(classes))
print("Number of pictures:", len(faces))

# Take smiling faces for eigenfaces
facematrix = []
facelabel = []
for key,val in faces.items():
    if key.startswith("ns1/"):
    continue
    if key.startswith("ns2/"):
    continue
    if key.startswith("ns3/"):
    continue
    if key.startswith("ns4/"):
    continue
    if key.startswith("ns5/"):
    continue
    if key.startswith("ns6/"):
    continue
```

```
        if key.startswith("ns7/"):
            continue
        if key.startswith("ns8/"):
            continue
        if key.startswith("ns9/"):
            continue
        if key.startswith("ns10/"):
            continue
        facematrix.append(val.flatten())
        facelabel.append(key.split("/")[0])

# Create facematrix as (n_samples,n_pixels) matrix
facematrix = np.array(facematrix)

# Apply PCA to extract eigenfaces
from sklearn.decomposition import PCA
pca = PCA().fit(facematrix)

print(pca.explained_variance_ratio_)

# Take the first K principal components as eigenfaces
n_components = 3
eigenfaces = pca.components_[:n_components]
```

```
# Show the k eigenfaces

fig, axes = plt.subplots(3,sharex=True,sharey=True,figsize=(8,10))

for i in range(3):

    axes[i].imshow(eigenfaces[i].reshape(faceshape), cmap="gray")

plt.show()


# Generate weights as a KxN matrix where K is the number of eigenfaces and N the number of samples

weights = eigenfaces @ (facematrix - pca.mean_).T

print("Shape of the weight matrix:", weights.shape)


# Test on out-of-sample image of existing class

query = faces["ns1/face1.pgm"].reshape(1,-1)

query_weight = eigenfaces @ (query - pca.mean_).T

print(query_weight.shape)

euclidean_distance = np.linalg.norm(weights - query_weight, axis=0)

print("Now to find best match !")

best_match = np.argmin(euclidean_distance)

print("Best match %s with Euclidean distance %f" % (facelabel[best_match], euclidean_distance[best_match]))

# Visualize

fig, axes = plt.subplots(1,2,sharex=True,sharey=True,figsize=(8,6))

axes[0].imshow(query.reshape(faceshape), cmap="gray")

axes[0].set_title("Query")

axes[1].imshow(facematrix[best_match].reshape(faceshape), cmap="gray")
```

```
axes[1].set_title("Best match")
plt.show()


# Visualize the mean face and random face
fig, axes = plt.subplots(1,2,sharex=True,sharey=True,figsize=(8,6))
axes[0].imshow(pca.mean_.reshape(faceshape), cmap="gray")
axes[0].set_title("Mean face")
random_weights = np.random.randn(n_components) * (weights.std())
newface = random_weights @ eigenfaces + pca.mean_
axes[1].imshow(new face.reshape(faceshape), cmap="gray")
axes[1].set_title("Random face")
plt.show()
```