Faculty of Engineering
Computer Engineering Department

Face Detection & Recognition Using Python

A Graduation Project Report
Presented in Partial Fulfillment of the Requirements for GEN001

Presented by

Name	Section	BN	Grade
Ahmed Mohamed Hassan Abdulrahman	4	25	
Ahmed Mostafa Abubakr	4	61	
Ahmed Mohamed Hamdy	4	27	
Ahmed Mohamed Sayed Awwad	4	29	
Ahmed Mohamed Sayed Noor ElDin	4	30	
Ahmed Mohamed Mahmoud Akrab	4	48	
Ahmed Mohamed Amin	4	21	

In partial fulfillment of the Requirements for
GEN001 Practical and Engineering Applications (Physics part)

Under supervision of: Dr. Nouran Mohamed

December 2018

Abstract

Face Recognition has been a hot topic for the past 20 years. This Paper Provides an up-to-date research about the sophisticated algorithms used for the recognition of faces. In simple words, a face must be detected (turned into a matrix so that a computer can read it) from a wide range of Images inputted into the database; using features like eyes, nose, mouth and other small details. Then, a machine learning recognizer is used to evaluate the database provided and start the recognition part of each person, this part is the trickiest part as some similar looking people may get the algorithm buzzed. Lastly, The main algorithm is used to detect and recognize multiple faces from a live video feed by extracting each frame and applying what it has learned from the pre-recognized faces in each frame and displaying it on the Graphical User Interface.

Contents

Abstract ii
Chapter 1: Introduction 1
1.1. Motivation and Justification 1
1.2. Problem Definition 1
1.3. Summary of Approach 2
Chapter 2: Literature Survey 2
Chapter 3: Necessary Background 5
Algorithms of Face Recognition 6
Types of Face Recognition algorithms 7
PCA (Principal Component Analysis) algorithm 7
ICA (Independent Component Analysis) Algorithm 8
LDA (Linear Discriminant Analysis) algorithm 8
Chapter 4: Design 10
Chapter 5: System Description 11
Chapter 6: Results 15
Chapter 7: Conclusion and Future Work 20
References 20

Chapter 1: Introduction

Face recognition systems are known to be critically dependent on manifold learning methods. A gray-scale face image of an order (A x B) can be represented as an AB-dimensional vector in the original image space. However, any attempt at recognition in such a high dimensional space is vulnerable to a variety of issues. Therefore, at the feature extraction stage, images are transformed to low-dimensional vectors in the face space. The main objective is to find such a basis function for this transformation which could distinguishably represent faces in the face space. A number of approaches have been reported, such as Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA). Primarily, these approaches are classified in two categories, i.e., reconstructive and discriminative methods. Reconstructive approaches (such as PCA and ICA) are reported to be robust for the problem of contaminated pixels, whereas discriminative approaches (such as LDA) are known to yield better results in clean conditions.

1.1. Motivation and Justification

Face Recognition is one of the systems that developed rapidly specially in the last few years. It takes part in a lot of fields like security systems, registering the attendance of employers, video game consoles…. etc.. Face recognition is one of the most safe security systems that is difficult to be tricked. Face recognition became like a field of competition between many mobile phone companies. It became one of the important features of mobile phones. Year after year they try to develop its accuracy and to reduce time that is needed to unlock the mobile phone. Face recognition is also useful for the disabled persons as it helps them using their mobiles. Also face recognition is used in cameras to determine face position to get good photos with good focus on people in the photo and some features in cameras like HDR are depending on face recognition systems. There are a lot of methods and algorithms used in manufacturing of face recognition systems and each of them has its own advantages and disadvantages and fields in which it is used. In the future face recognition may replace many jobs as security men and governmental clerks. Face recognition is also used in cars. Byton is a car manufacturing company that now uses face recognition systems to unlock doors instead of keys.

1.2. Problem Definition

The aim of our research is to develop a theoretical model and a set of terms for understanding and discussing how we recognize familiar faces, and the relationship between recognition and other aspects of face processing. It is suggested that there are seven distinct types of information that we derive from seen faces; these are labelled pictorial, structural, visually derived semantic, identity-specific semantic, name, expression and facial speech codes. A functional model is proposed in which structural encoding processes provide descriptions suitable for the analysis of facial speech, for analysis of expression and for face recognition units. Recognition of familiar faces involves a match between the products of structural encoding and previously stored structural codes describing the appearance of familiar faces, held in face recognition units. Identity-specific semantic codes are then accessed from person identity nodes, and subsequently name codes are retrieved. It is also proposed that the cognitive system plays an active role in deciding whether or not the initial match is sufficiently close to indicate true recognition or merely a ‘resemblance’; several factors are seen as influencing such decisions. And to achieve robust performance, we use the model to con- strain solutions to be valid examples of faces. A model also provides the basis for a broad range of applications by 'explaining' the appearance of a given image in terms of a compact set of model parameters, which may be used to characterize the pose, expression or identity of a face. In order to interpret a new image, an efficient method of finding the best match between image and model is required.

1.3. Summary of Approach

As we looked into the project, we think the best approach to develop a Face Recognition System is by using special libraries like OpenCV and implement the algorithms (PCA, ICA, LDA…etc) by programming means and languages like Python or C++, as they provide a better representation of Face Recognition Systems. It will yield results we can depend on. So in our live camera face recognition system we are using **OpenCV** with **Haar Cascade Classifier** for detecting faces, **Python** as a programming language and **PCA**; which is the algorithm we depend on.

Chapter 2: Literature Survey

2.1: Principle Component Analysis

	UMSIT	ORL	YALE
MPCA	63.00 (10/09)	94.44 (25/04)	94.44 (25/09)

Those authors that have used PCA for face recognition (such as *Jose* and *Gottumukkal et al*) proposed the local and global feature extraction method by exploiting Modular Principal Component Analysis (MPCA) that incorporates Modular PCA and 2D PCA method. The algorithm deals with N-smaller subdivision of faces in order to increase its recognition rate by taking the benefit of un-harmful regions. Consequently average image is computed from N-sub images. On the whole the results are accomplished on Yale, ORL and UMIST database under the un-controllable conditions of face recognition like pose, illumination and expression variation and attain the results as shown in Table 1.

Table 1. Best Recognition Rates

Nicholl and Amira, focused on automatically estimating the discriminative coefficients in a DWT/PCA that deals with inter-class and intra class standard deviations. Consequently, Eigen-faces are elected on the basis of Eigen-values with discrepancy due to the illumination factors between trained images. Recognition rate of such system is shown in Table 2.

Table 2. Comparative Results

	Accuracy (%)
DWT/PCA with Coefficient selection	96.5

Li et al gained an attention to feature extraction through PCA for feature extraction and LDA for classifying features. Similarly, Nearest Neighbor Classifier (NNC) is used for face recognition. Table 3 demonstrates the outcome

produced by the approach in ORL database.

Table 3. Recognition Rate

	5	**10**	**15**	**20**	**25**	**8**	**35**	**40**
PCA + LDA	0.635	0.78	0.845	0.835	0.89	0.895	0.9	0.93

Moon and Philips, accessed and analyzed different PCA based algorithms on the basis of comparison with illumination normalization; upshots of compressed images varied Eigen-vectors illustration and alter the similarity measure in feature classification methods. *Chen et al*, on the other hand proposed Adaptive Principal Component Analysis (APCA) in order to progress the PCA results by functioning PCA on faces, rotate face space and then warping is applied between class and within class covariance. *Zhang et al*, presents subspace method so called Diagonal Principal Component Analysis (DiaPCA) that deal with superlative projective vectors and conclude the results that the proposed algorithm provides more accuracy in contrast to PCA and 2DPCA. *Hongta*, proposed an idea in a new direction by adapting multi feature extraction approach that incorporates PCA and LDA. On the other hand, exploit Radius Basis Function Network (RBFN) for feature classification. In [1] PCA based face recognition is outlined which is varied in terms of testing criteria. The recognition rates vary depending on the number of training and testing sets used size of the image and even presence of noise in the face images. In [2] PCA based reconstruction procedure is applied with its novelty in using the median vector rather than the average of the class samples. It has the inherit advantage of preserving the spatial information of image samples as well as its robust nature towards outliers. In [3], a Particle Swarm Optimization (PSO) is adopted as a new method for selecting important face features. This variation of selecting the discriminatory feature instead of PCA showed great performance in terms of recognition rates. A combination of wavelet transform and SVM methodology is discussed in [4]. The former is used as a preprocessing technique while later is for classifying the facial features. A modular PCA based face recognition system is illustrated in [5], in combination with within-class median. For all the training and testing images normalization of sub images samples is performed before projection. This result in increased recognition performance in contrast to other modular based PCA based methods.

2.2: Independent Component Analysis

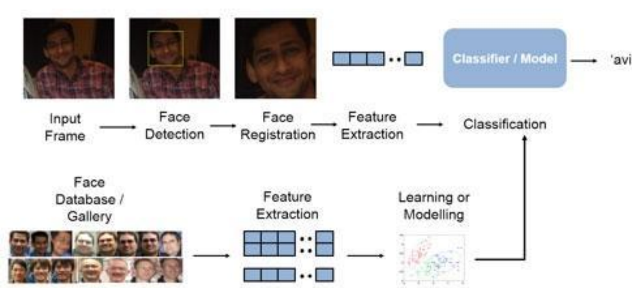
The calculation of ICA was discussed in several recent papers [6, 7, 8, 9, 10, 11], where the problem was given various names. For instance, the terminology “sources separation problem” has often been coined. Investigations reveal that the problem of 'independent component analysis' was actually first proposed and so named by *Herault and Jutten* around 1986 because of its similarities with principal component analysis (PCA). This terminology is retained in the paper. *Herault and Jutten* seem to be the first (around 1983) to have addressed the problem of ICA. Several papers by these authors propose an iterative real-time algorithm based on a neuro-mimetic architecture. The authors deserve merit in their achievement of an algorithmic solution when no theoretical explanation was available at that time. Nevertheless, their solution can show lack of convergence in a number of cases. Refer to [10] and other papers in the same issue, and to [12]. In their framework, high-order statistics were not introduced explicitly. It is less well known that Bar-Ness [13] independently proposed another approach at the same time that presented rather similar qualities and drawbacks. *Giannakis* et al. [14] addressed the issue of identifitiability of ICA in 1987 in a somewhat different framework, using third-order cumulants. However, the resulting algorithm required an exhaustive search. *Lacoume and Ruiz* [15] also sketched a mathematical approach to the problem using high-order statistics; in [16, 8], *Gaeta and Lacoume* proposed to estimate the mixing matrix M by maximum likelihood approach, where an exhaustive search was also necessary to determine the absolute maximum of an objective function of many variables. Thus, from a practical point of view, this was realistic only in the two-dimensional case. Cardoso focused on the algebraic properties of the fourth-order cumulants, and interpreted them as linear operators acting on matrices. A simple case is the action on the identity yielding a cumulant matrix whose diagonalization gives an estimate of ICA [6]. When the action is defined on a set of matrices, one obtains several cumulant matrices whose joint diagonalization provides more robust estimates [17]. This is equivalent to optimizing a cumulant-based criterion [17], and is then similar in spirit to the approach presented herein. Other algebraic approaches, using only fourth-order cumulants, have also been investigated [18, 19]. In [9], Inouye proposed a solution for the separation of two sources, whereas at the same time *Comon* [20] proposed another solution for N 7>= 2. Together with *Cardoso's* solution, these were among the first direct (within polynomial time) solutions to the ICA problem. In [11], *Inouye* and his colleagues derived identifitiability conditions for the ICA problem. Their Theorem 2 may be seen to have connections to earlier works [21]. On the other hand, our Theorem 11 (also presented in [22, 7]) only requires pairwise independence, which is generally weaker than the conditions required by [11]. In [23], *Fety* addressed the problem of identifying the dynamic model of the form y(t) = Fz(t), where t is a time index. In general, the identification of these models can be completed with the help of second-order moments only, and will be called the signal separation problem. In some cases, identifitiability conditions are not fulfilled, e.g. when processes z,(t) have spectra proportional to each other, so that the signal separation problem degenerates into the ICA problem, where the time coherency is ignored. Independently, the signal separation problem has also been addressed more recently by *Tong et al*. [24]. This paper is an extended version of the conference paper presented at the Chamrousse workshop in July 1991 [7]. Although most of the results were already tackled in [7], the proofs were very shortened, or not stated at all, for reasons of space. They are now detailed here, within the same framework. Furthermore, some new results are stated, and complementary simulations are presented. Among other things, it is sought to give sound justification to the choice of the objective function to be maximized. The results obtained turn out to be consistent with what was heuristically proposed in [15]. Independently of [16, 8], the author proposed to approximate the probability density by its Edgeworth expansion [7], which has advantages over *Gram-Charlier's* expansion. Contrary to [16, 8], the hypothesis that odd cumulants vanish is not assumed here: *Gaeta* emphasized in [16] the consistency of the theoretical results obtained in both approaches when third-order cumulants are null. It may be considered, however, that a key contribution of this paper consists of providing a practical algorithm that does not require an exhaustive search, but can be executed within a polynomial time, even in the presence of non-Gaussian noise. The complexity aspect is of great interest when the dimension of y is large (at least greater than 2). On the other hand, the robustness in the presence of non-Gaussian noise has not apparently been previously investigated. Various implementations of this algorithm are discussed in [25, 26, 22]; the present improved version should, however, be preferred. A real-time version can also be implemented on a parallel architecture [26].

Chapter 3: Necessary Background

In this paper we are going show you some things that you need to know about face recognition technology that is our project is about.

Face recognition is a type of biometric system which is a technological system that uses information about a person and his biological characteristics to identify that person and there are many applications to this technology like (Fingerprint, Face Recognition, and Iris Recognition ….etc). So we can conclude that the purpose of facial recognition is identifying people.

Face Recognition is a technique uses certain algorithms to detect faces of individuals using pre-saved data of digital images using certain types of software like openCV, algorithms as PCA, LDA, SVM …etc. and programming languages as Python, Octave …etc, by analyzing pattern of images or videos and comparing it with its database as shown in fig(1). The first face recognition system was created in the 1960s and

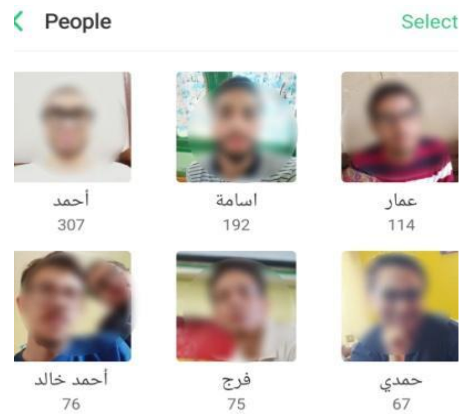


Fig(1) "face classification example"

it was developing year after year and its accuracy increased so much that some software uses 3D characteristics of faces and some face recognition system now is able to recognize animals' faces

The accuracy of Face Recognition is less than the accuracy of other techniques as fingerprint and voiceprint but it provides many facilities and it is used in many applications (Security, Organization of data as shown in fig(2), Finding wanted criminals, … etc).
So we need to start a face recognition system:
-A database of images.
-A software to perform the system on it
-Some codes to use in the software
-An image to test the system efficiency

<u>So you ask yourself now how we can make such good technology?!</u>



Fig(2) image from oppo mobile phone showing an example for data classification

Well Face recognition is based on 3 main operations:

1- Having a database.
2- Import an image to be tested.
3- Test the image to know if it from the database or not.

We will talk about them in some details.

1- Having a database:

It is simply like the password that you put in your mobile phone to keep your information secured, database is a collection of images of people that you want to access to your information, but to make the program read these images they should have some properties:

- Having the same dimensions (size) so the program can read it all in the same scale.

- Consider that the images contain different positions of the face to increase the accuracy of the recognition

- A good quality image with suitable brightness so that the person's face is obvious in it, as the program uses face characteristics in the recognition process.

- A solo person image to avoid any mistakes in reading image by the system.

<u>But you may ask yourself how programs see images ?</u>

Well Programs see images as pixels which are very small elements that the images are made from. It's colored gradually from black(0) to white (255) and through these numbers you can conclude that the program will not see images with colors, but as numbers, or in other words, a **matrix** which is a two-dimensional matrix that the program reads and then saves to the database.

Now the program has some database, but it needs to know how to use it and here comes the coding operation.

So, to use face recognition you need to have a reliable software and to know some codes, this simply can be compared to a building site where there is the software and the engineer and codes are the structure materials, and hence we are going to save our database and use it.

Now you have all the files needed to start coding operations and in our project we are going to use openCV as our Library/ software and python as the coding language. We start by installing openCV and python and then through some codes, we import the image of the database and then save it. The codes' role here is to do some operations or functions to the images and by getting some codes together to perform a specific function is simply called an algorithm

Algorithms of Face Recognition

Algorithms are the steps of solving a problem or achieving an aim. So if we consider codes as the units that build the system of face recognition then the algorithms are the steps of writing these codes and their order also it specify the type of the used codes and their functions. There are different types of face recognition algorithms each of them have advantages and disadvantages. Each algorithm has its accuracy but accuracy is not the only thing we should consider while choosing the type of algorithm as each algorithm has its own conditions that it performs quite perfectly compared to any other type of algorithms.

Types of Face Recognition algorithms

First- Geometric (Template) based algorithms

These types of algorithms are based on analyzing the geometric relations between local facial features. It contains many types of algorithms like PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) and SVM (Support Vector Machines).

Second- Piecemeal
This type of algorithms is based on deducing the most relevant characteristics such as eyes or some combinations of face characteristics.

And now you are about to take a look on some examples of the first type of algorithms (Geometric based algorithms).

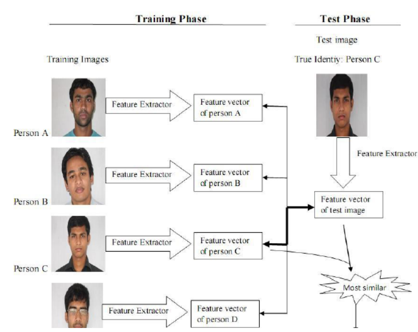
PCA (Principal Component Analysis) algorithm

PCA is a statistical procedure which is one of the most used algorithms in face recognition systems and it depends on the geometric relations between face characteristics. It was invented in 1901 by Karl Pearson and it was developed later in 1930s by Harold Hotelling. The simplest and most effective approach of PCA is called Eigenfaces which convert faces into small sets of characteristics. Recognition is then done by inserting the image needed into the Eigenface subspace then the person is classified by comparing its position with the positions of images of individuals in the saved database. The advantage of Eigenface is in its speed and its sensitivity to small and gradual changes. Its accuracy ranges between 61.25-92.5%.

The recognition process in Eigenface algorithm is involved in **following steps**:

Initialization process: in this step the initial sets of face images is measured and called training sets then from these sets the Eigenfaces are calculated and the highest eigenvalues are kept after detecting the space of the face in images. The distribution is calculated by projecting faces of persons in database into the face space

Recognition process: Inserting the input image into each of the eigenfaces, then determine if the image is face or not after this by classifying the patterns it deduce if the image is for a known person from the database or not as shown in figure (3). If images of certain unknown person are used several times it is added to database as a known person



Fig(3) Example of recognition of image

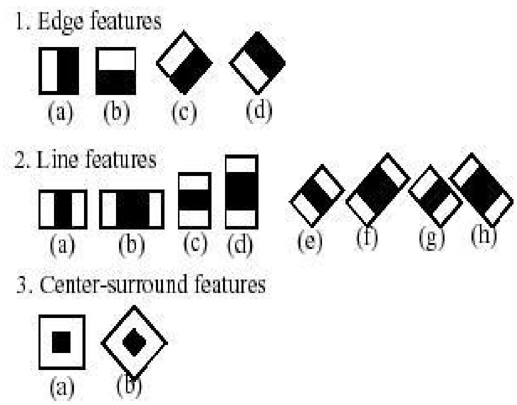
ICA (Independent Component Analysis) Algorithm

Despite the advantages of PCA but it lacks information at high statistics orders. ICA provides more powerful data representation than PCA as it doesn't depend highly on statistics, hence ICA yields better results. ICA in face recognition depends on the distance between face characteristics and their size also it analyzes their depth and geometric shape. It's not as commonly used as PCA or LDA

LDA (Linear Discriminant Analysis) algorithm

LDA algorithm is dealing with faces in its database as classes and it analyzes the input face then it puts the images of the same person in the same class. Images are represented as a large number of pixels. LDA reduces the number of features to a more manageable number before recognition. The average accuracy of LDA face recognition is 90.8%.

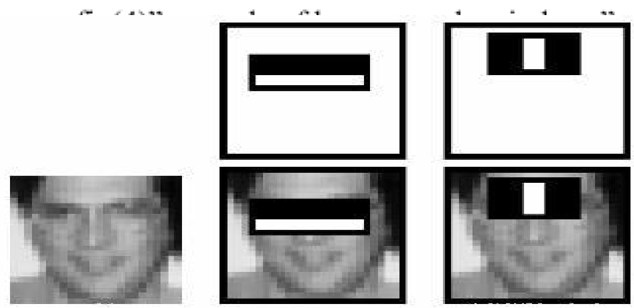
So when image is imported in face recognition system it is affected by one of these algorithms or others to be put in database or to be tested in the system.



CV2

CV2 is a library that gives us permissions needed to operate the System like using the camera to use live video during recognition process.

Haar Cascade



fig(5) "extracting features by windows"

Haar Cascade is one of the modules we used in our project. It's responsible for the first stage of the process which is face detection. It gives us the ability to detect faces from images by using positive (with face) and negative (without face) pictures in its database. Haar Cascade puts windows on haar features as shown in fig(4). Every feature has a single value obtained by subtracting the number of pixels under the white region of the window from the black one. The black region and the white one depend on the degree of color and brightness of inserted images. All possible sizes of each window are placed on all possible locations of each image to calculate plenty of features. Fig(5) shows an example of extracting features of eyes and nose region. The black part of the window is on the eyes as it darker than the nose and the cheeks. Every face detection system is described by its accuracy and speed in haar cascade the accuracy is high but its speed is not. So if we are talking about Accuracy haar cascade is good option to choose. It is preferred also to use pictures and videos of good brightness as haar cascade is very sensitive to light and its accuracy decreases on dark pictures.

Numpy

Numpy is library that is responsible for all the mathematical operations we will use in importing the database and reading images.

Grayscale

Images must be converted to gray scale so that all colors are converted to degrees of black to simplify the images hence it will be easily converted to numerical values that will be lately be used in recognition process.

And this is how a face recognition system is done all the left now is testing it by importing two images one from a person in the database and other not and see how the system reacted to these tests :)

Fig(6) “Example from our face recognition system

Finally in Fig(6) a picture taken from and Python.

Chapter 4: Design

To achieve face recognition, the used program needs to:

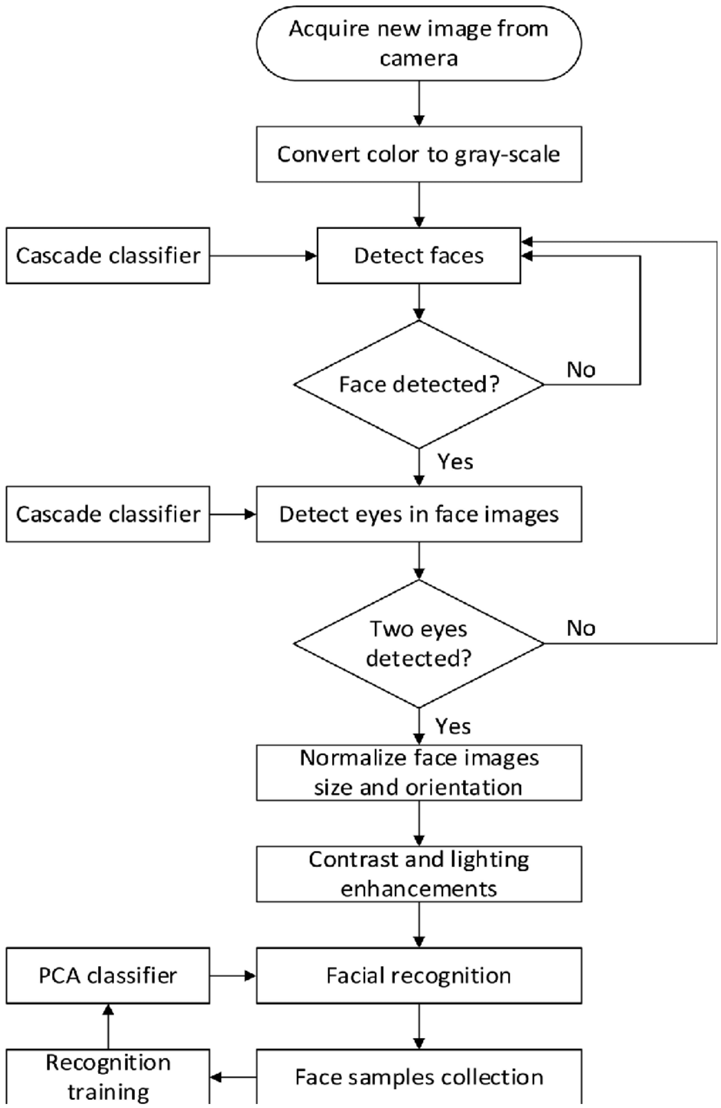
1-Identitfy face shapes through other objects.

2-Mark every face landmarks through other faces.

3-Save this information as database.

4-use the data again to identify faces.

5-finally getting true if the information applies he used face and getting false if it not.



fig(8): Flow chart

Chapter 5: System Description

first to get the system to identify a face from other shapes we used haar_cascade Classifier and specifically haarcascade_frontalface_alt it excels at catching face landmarks throughout pictures and videos and it can be defined as the difference of the sum of pixels of areas inside a rectangle, which can be at any position and scale within the original image. Hence, by trying to match each feature (at different scales) in the database with different positions in the original image, the existence or absence of certain characteristics at the image position can be obtained. These characteristics can be, for example; edges or changes in textures. Hence, when applying a set of haar-like features pre-trained to match certain characteristics of facial features, the correlation by which a certain feature matches an image feature can tell something about the existence or non-existence of certain facial characteristics at a certain position. As an example, in the following figure a haar-like feature that looks similar to the bridge of the nose is applied onto a face. One of the contributions of Viola and Jones was to use summed area which they called Integral images can be defined as two-dimensional lookup table in the form of a matrix with the same size of the original image. Each element of the integral image contains the sum of all pixels located on the up-left region of the original image (in relation to the element's position). This allows to compute sum of rectangular areas in the image, at any position or scale, using only four lookups:

Where points belong to the integral image , as shown in the figure. Each Haar-like feature may need more than four lookups, depending on how it was defined. Viola and Jones's 2-rectangle features need six lookups, 3-rectangle features need eight lookups, and 4-rectangle features need nine lookups.

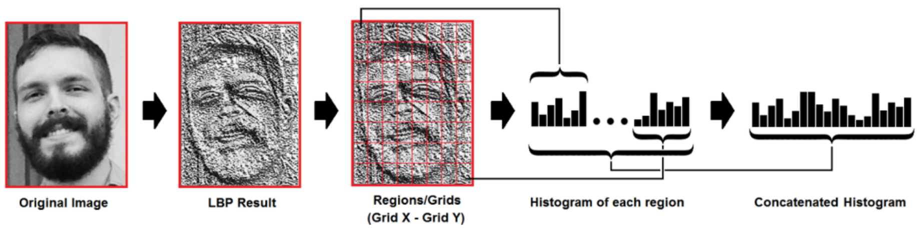
To use Haar cascades in the system we use python
With openCV library fruiters in this code we are going to use Cv2, Numpy, and pickle
Following this code in python:
>>import cv2
>>import Numpy as np
>>import pickle
#these futures, from the OpenCV library has the all the algorithms and function needed to start capturing faces.
>>face_cascade= cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_alt.xml')
#using one of cv2 functions (cascade_frontalface_alt.xml) which was mentioned earlier
>>recognizer=cv2.face.LBPHFaceRecognizer_create() #cv2 is providing a very important algorithm in this line which is Local Binary Patterns Histograms (LBPH).it is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

The LBPH uses 4 parameters:

- **Radius**: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- **Neighbors**: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- **Grid X**: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

- **Grid Y**: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

Using the image generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, as can be seen in fig(7):



Fig(7) turning image though its pixels to histogram.

As we have an image in gray scale, each histogram (from each grid) will contain only 256 black color intensities (0~255) representing the occurrences of each pixel intensity.

Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

- In face recognition LBPH is use various approaches to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance**, **chi-square**, **absolute value**, etc. In this based on the following formula:

- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a '**confidence**' measurement. **Note**: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.
- We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

So after that name this algorithm any name like recognizer.

>> Recognizer = cv2.face.LBPHFaceRecognizer_create()

Now we have the algorithm ready, so to import the database we are going to follow these codes:

>> current_id = 0

#to recognize the first id and its information

>> label_ids = {}

#to understand this line we have to understand the meaning of the label, which is in the programming language, that is a sequence of characters that identifies a location within source code. In most languages labels take the form of an identifier, often followed by a punctuation character (e.g., a colon). In many high level programming languages the purpose of a label is to act as the destination of its statement. In assembly language labels can be used anywhere an address can, so this code identifies ids locations to use it.

>> y_labels = []

#our data is going to be represented in the y label which we going to use later in the loop

>> x_train = []

Now we are going to use a loop to link the data with the algorithm…..

```
>> for root, dirs, files in os.walk(image_dir):
    For file in files:
            If file.endswith("png") or file.endswith("jpg"):
                Path = os.path.join(root, file)
```

#here the code's purpose is to identify the file's database type which of course is image type png or jpg which is the only file that the program should cheek on. Then os.path contains functions for manipulating filenames and directory names. The os.path.join() function constructs a pathname out of one or more partial pathnames. In this case, it simply concatenates strings. Calling the os.path.join() function will add an extra slash to the pathname before joining it to the filename.

```
    Label = os.path.basename(root).replace(" ", "-").lower()
                    If not label in label_ids:
                            label_ids[label] = current_id
                            current_id += 1
                    id_ = label_ids[label]
y_labels.append(label)      #some number
```

x_train.append(path)   # verifies this image and turn Numpy array, then turn it into a gray scale image pil_image = Image.open(path).convert("L")   #gray scale

Size = (550, 550)

Final_image = pil_image.resize(size, Image.ANTIALIAS)

image_array = np.array(pil_image, "uint8")
#turning it into a Numpy array

Faces = face_cascade.detectMultiScale(image_array,
ScaleFactor=1.2, minNeighbors=5)

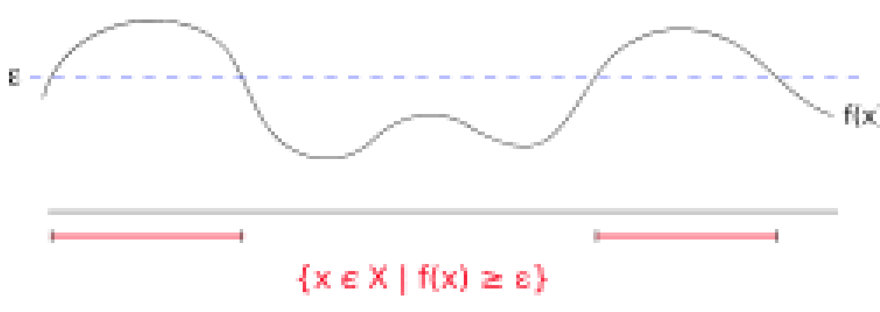
Face_cascade.detectMultiScale is one of OpenCV tools which detects objects which are different in the input image, the detected objects are then processed as a list of rectangles.

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

#which are the directions of any face or image so the algorithm can save it though these information

Roi = image_array[y:y+h, x:x+w]

The Region of Interest (ROI) is a function that enables a non-uniform distribution of the image quality between a selected region (the ROI) and the rest of the image (background).



frame

Chapter 6: Results

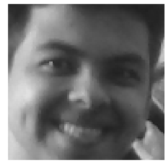
Despite all the extensive research effort that has gone into computational face recognition algorithm, We have yet to see a system that can be deployed effectively in an unconstrained setting, with all of the attendant variability in imagining parameters such as sensor noise, viewing distance ,and illumination. The only system that does seem to work well in the face of these challenges is the human visual system.

We have noticed that the system cannot recognize faces in very low-resolution image, although humans can recognize familiar faces in very low-resolution images. Progressive improvements in camera resolution provide ever-greater temptation to use increasing amount of details in face representations in machine vision system. Higher image resolutions allow recognition system to discriminate between individuals on basis of fine differences in their facial features.
We have built our system by using the Haar cascade classifier over LBP classifier, Both of these classifier process image in gray scale.
Haar cascade: Is a machine learning based approach, an algorithm created by Paul viola and Michael jones; which are trained from many positive images (with faces) and negative images (without faces).
LBP classifier: As any other classifier, the Local Binary patterns or LBP, also needs to be trained on hundreds of images, LBP is a visual/texture descriptor, LBP features are extracted to form a feature vector that classifies a face from a non-face.
Each openCV face detection classifier has its pros and con, but the major differences are in accuracy and speed, despite the many disadvantages of the haar cascade, we have chosen the haar cascade for the high detection accuracy.
In Addison to haar cascade classifier we have decided to build the system with LBPH method (Local Binary patterns Histograms), and it's a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixels and considers the results as binary number.

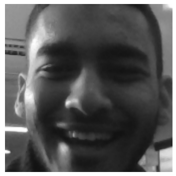
Algorithm	Advantages	Disadvantages
Haar	1. High detection accuracy 2. Low false positive rate	1. Computationally complex and slow 2. Longer training time 3. Less accurate on black faces 4. Limitations in difficult lightening conditions 5. Less robust to occlusion
LBP	1. Computationally simple and fast 2. Shorter training time 3. Robust to local illumination changes 4. Robust to occlusion	1. Less accurate 2. High false positive rate

Using the LBP with histograms we can represent the face images with a simple data vector, and it can also be used for face recognition tasks.
We have also noticed some demerits in our system, one of these demerits is the system cannot recognize faces if there is any defects in the images of the data base of that person, and I means by defects such as unequal distributed light in the photo (Illumination defects) .if there are any illumination defects in the data base, as a result, the haar cascade will no detect the face in the images, Because the system transform the image into a gray scale.
The collected images are shown below. Each face has 25 images. Two applications were written to iterate through the parameters of each algorithm. On each literation, the algorithm is trained using different parameters and tested against a photo. The resulting data is plotted at the after finishing the tests.
The applications are:
Test Data Collector EigenFace.py.
Test Data Collector LBPH.py.






ID:1 ID:2 ID:3 ID:4

ID:5 ID:6 ID:7 ID:8

The first test image is shown in figure 5 and the plots are analyzed below. The resulting ID change is plotted below in figure6. Note when components were 1, it identified

Figure 1: Image used for this test

The face as ID-6 and the rest are between ID-7 and ID-8, which is the same person.

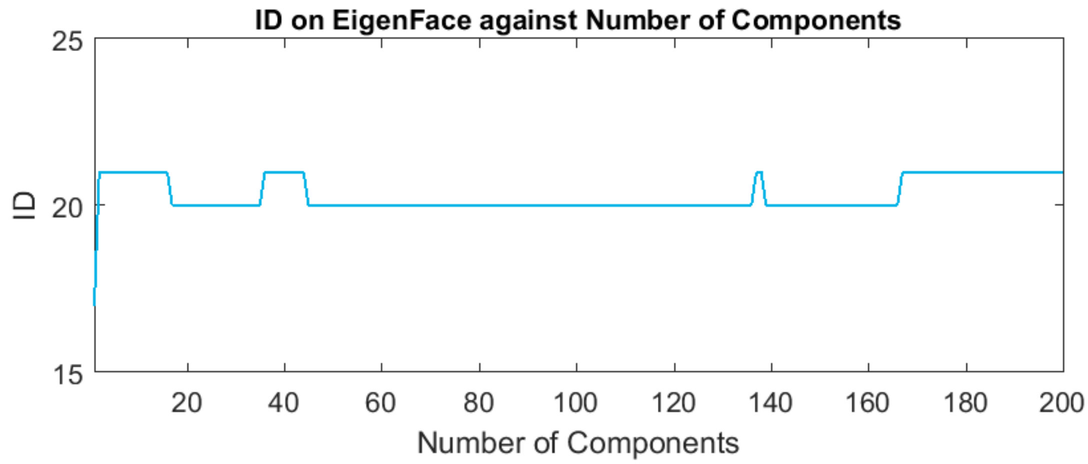


Figure 2: The ID from the face recognizer changes between two classes of the same person.

Confidence is plotted in figure 3, increasing with components. From this plot it appears the best is when components are below 20.

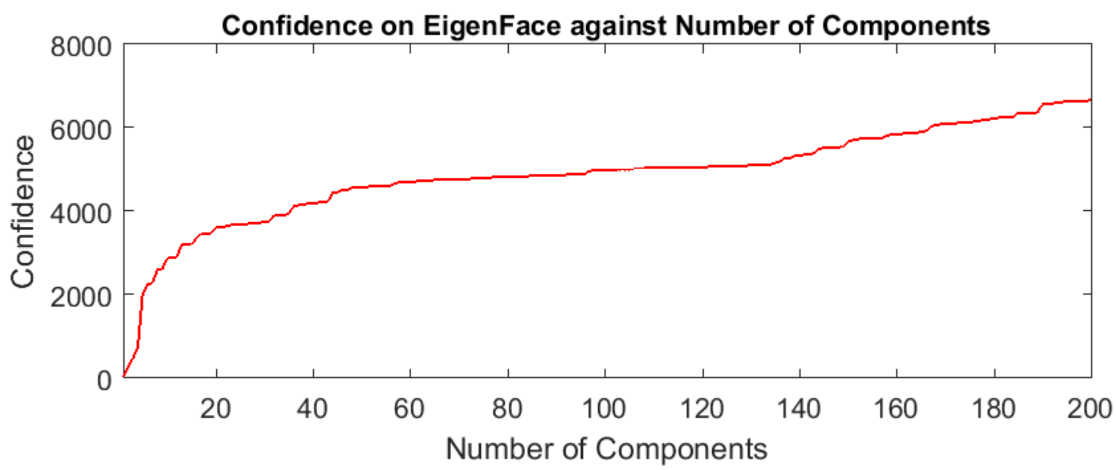


Figure 3: The Confidence increasing with No. of components

.
LBPH has more than one parameter to change. All are incremented to the maximum.

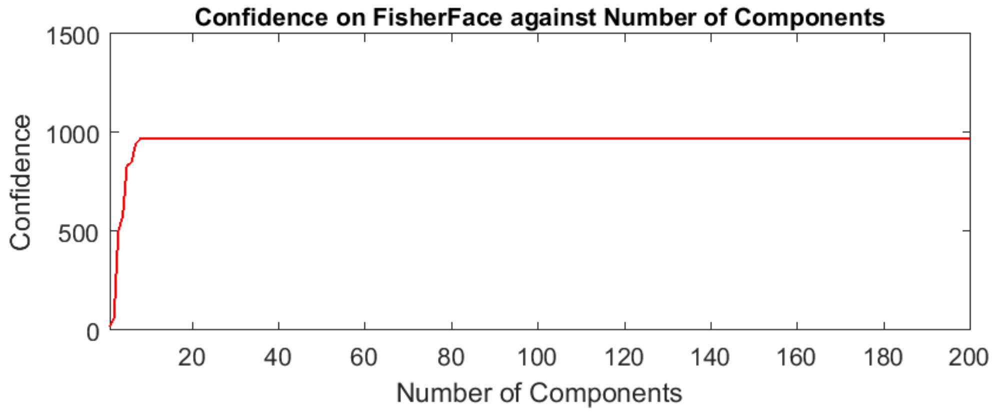
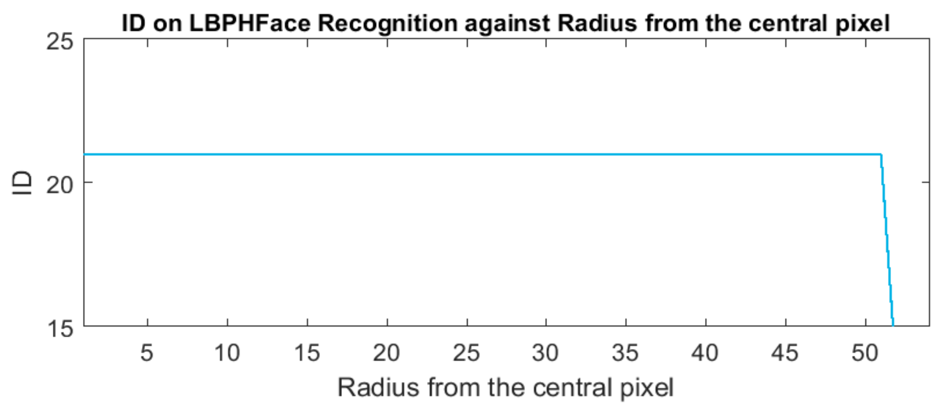


Figure 4: Stable confidence after 10 components

Limit and the results are shown below. The first is the radius from the center pixel and since the image size is 110 X 110, maximum radius is 54. The ID is steady



all the way to 50 as can be seen in figure 5

Figure 5: The ID returned from LBPH

Confidence level is graphed against the radius in figure 6. The confidence is fluctuating after 40.

The lowest confidence level is at 2

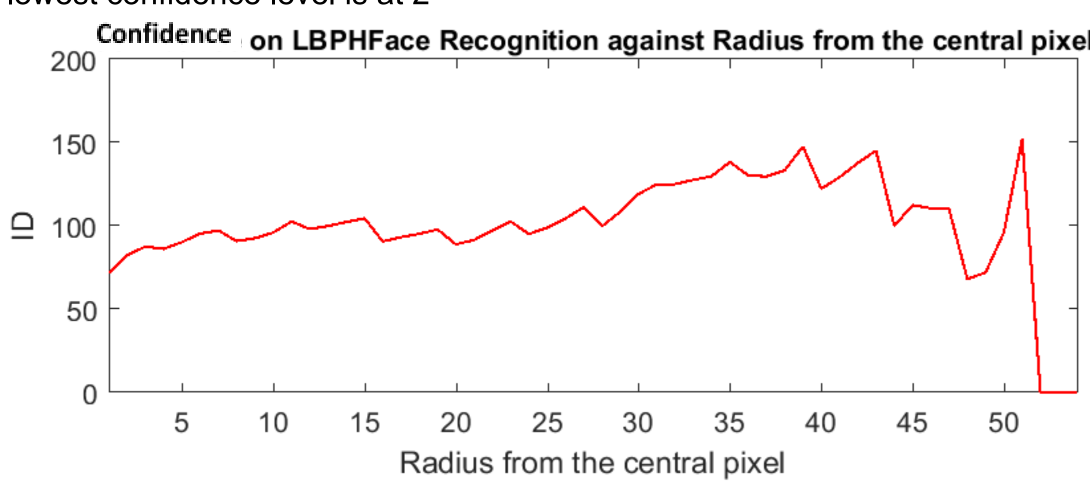


Figure 6: The confidence returned from LBPH

The number of neighbors are changed from 1 - 13. Further increase caused the computer to stall. The returned ID is plotted below in figure 7. ID steady until 9 neighbors and changed to ID-7.

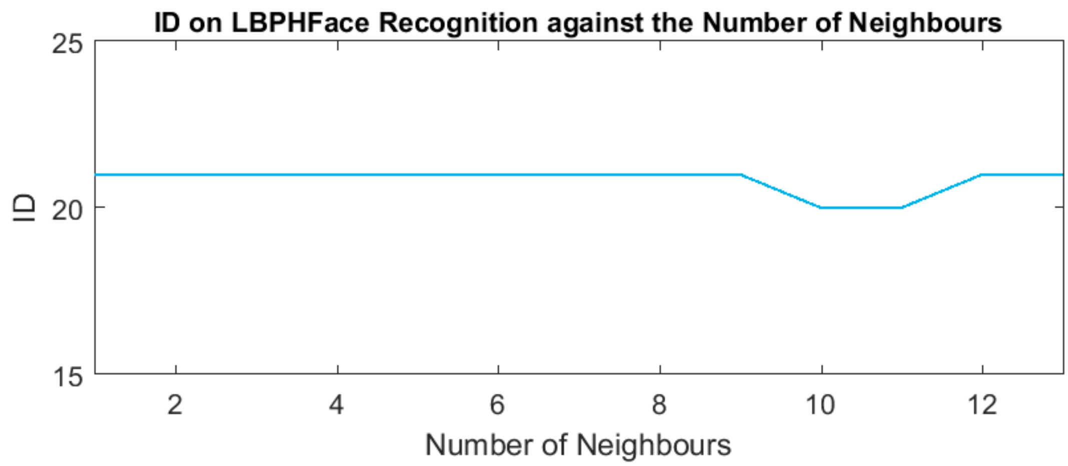


Figure 7: The ID returned from LBPH changing neighbor

The confidence continuously increased as can be seen in figure 20 and 1 neighbor will be included to the next test.

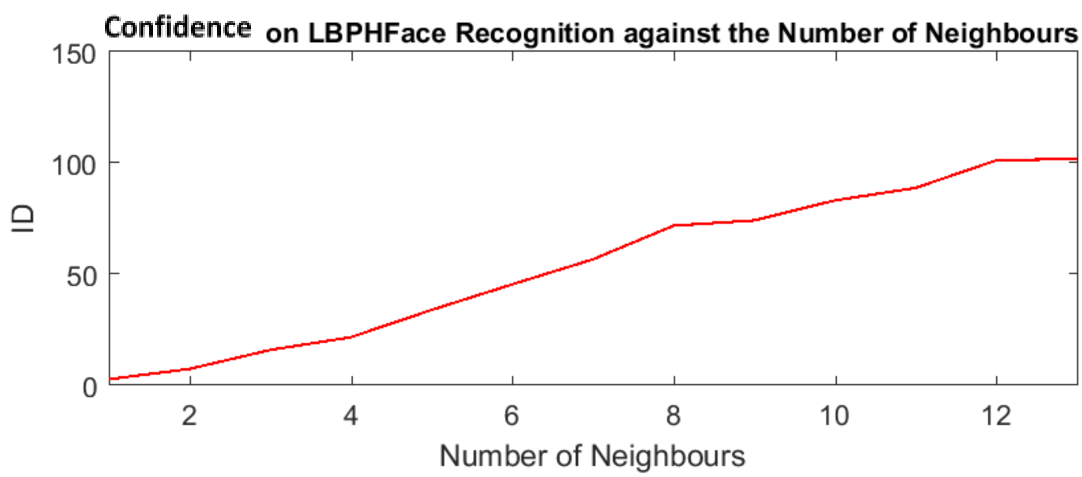


Figure 8: The Confidence returned from LBPH changing neighbors

Chapter 7: Conclusion and Future Work

We found the OpenCV approach astonishing so we plan to dig deeper into it and give more powerful and efficient languages like C++ a chance to implement more features such as adding a well- designed GUI, dramatically improving the efficiency by making the database entry with live video recording frame capture, and if we were successful in the Windows platform we will look forward to developed a cross platform version of our project on other platforms like Android, IOS, Linux.

References

[1] " Poon B., Amin M., and Yan H., "PCA Based Face Recognition and Testing Criteria," in Proceedings of the 8th International Conference on Machine Learning and Cybernetics, Baoding, vol. 5, pp. 2945-2949, 2009. "
[2] "Sharif M., Mohsin S., and Javed M., "Real Time Face Detection using Skin Detection (Block Approach)," Journal of Applied Computer Science and Mathematics, vol. 5, no. 10, pp. 75- 81, 2011."
[3] "Zhao W., "Discriminant Component Analysis for Face Recognition," in Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, vol. 2, pp. 818-821, 2000."
[4] "Wei X., Sheng Y., Qi W., and Ming L., "Face Recognition Based on Wavelet Transform and PCA," in Proceedings of Pacific-Asia Conference on Knowledge Engineering and Software Engineering, Shenzhen, pp. 136-138, 2009"
[5] "Xiao-Jie W., "Modular PCA Based on WithinClass Median for Face Recognition," in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, vol. 1, pp. 52- 56, 2010"
[6] J.F.Cardoso, "Sources separation using higher order moments", Proc. Internat. Conf. Acoust. Speech Signal PROCESS. Glasgow, 1989, pp. 2109-2112.
[7] P. Comon, "Independent component analysis", Internat. Sional Processin9 Workshop on High-Order Statistics, Chamrousse, France, 10-12 July 1991, pp. 111-120; republished in J.L. Lacoume, ed., Hioher-Order Statistics, Elsevier, Amsterdam 1992, pp. 29-38.
[8] M. Gaeta and J.L. Lacoume, "Source separation without a priori knowPedge: The maximum likelihood solution", in: Torres, Masgrau and Lagunas, eds., Proc. EUSIPCO Con.['., Barcelona, Elsevier, Amsterdam, 1990, pp. 621 624.
[9] Y. Inouye and T. Matsui, "Cumulant based parameter estimation of linear systems", Proc. Workshop HigherOrder Spectral Analysis, Vail, Colorado, June 1989, pp. 180-185.

[10] C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive al'gorithm based on neuromimatic architecture", Signal Processing, Vol. 24, No. 1, July 1991, pp. 1 10.
[11] L. Tong et al., "A necessary and sufficient condition of blind identification", Internat. Signal Processing Workshop on High-Order Statistic.s, Chamrousse, France, 10 12 July 1991, pp. 261-264.
[12] J.C. Fort, "'Stability of the JH sources separation algorithm", Traitement du Signal, Vol. 8, No. 1, 1991, pp. 35 42.
[13] Y. Bar-Ness, "Bootstrapping adaptive interference cancelers: Some practical limitations", The Globecom Conf., Miami, November 1982, Paper F3.7, pp. 1251-1255.
[14] G. Giannakis, Y. Inouye and J.M. Mendel, "Cumulantbased identification of multichannel moving average models", IEEE Automat. Control, Vol. 34, July 1989, pp. 783- 787.
[15] J.L. Lacoume and P. Ruiz, "'Extraction of independent components from correlated inputs, A solution based on cumulants", Proc Workshop Higher-Order Spectral Analysis, Vail, Colorado, June 1989, pp. 146-151.
[16] M. Gaeta, Les Statistiques d'Ordre Sup6rieur Appliqu6es :i la S6paration de Sources, Ph.D. Thesis, Universit+ de Grenoble, July 1991.
[17] A. Souloumiac and J.F. Cardoso, "Comparaisons de M&hodes de S6paration de Sources", XIIIth Coll. GRETSI, September 1991, pp. 661 664.
[18] J.F. Cardoso, "Super-symmetric decomposition of the fourth-order cumulant tensor, blind identification of more sources than sensors", Proc. Internat. Conj. Acoust. Speech Signal Process. 91, 14 17 May 1991.
[19] J.F. Cardoso, "Iterative techniques for blind sources separation using only fourth order cumulants", Conf. EUSIPCO, 1992, pp. 739 742.
[20] P. Comon, "Separation of stochastic processes", Proc. Workshop Higher-Order Spectral Analysis, Vail, Colorado, June 1989, pp. 174 179.
[21] G. Darmois, "Analyse G6n6rale des Liaisons Stochastiques", Rev. Inst. Internat. Stat., Vol. 21, 1953, pp. 2-8.
[22] P. Comon, "Analyse en Composantes Ind6pendantes et Identification Aveugle", Traitement du Signal, Vol. 7, No. 5, December 1990, pp. 435 450.
[23] L. Fety, Methodes de traitement d'antenne adapt+es aux radiocommunications, Doctorate Thesis, ENST, 1988.
[24] L. Tong, V.C. Soon and R. Liu, "AMUSE: A new blind identification algorithm", Proc. Internat. Conf. Acoust. Speech Signal Process., 1990, pp. 1783-1787.
[25] P. Comon, "Separation of sources using high-order cumulants", SPIE Conf. on Advanced Algorithms and Architectures for Siynal Processin9, Vol. Real-Time Signal Processin 9 XII, San Diego, 8 10 August 1989, pp. 170 181.
[26] P. Comon, "Process and device for real-time signals separation", Patent registered for Thomson-Sintra, No. 9000436, December 1989.
[27]**https://en.wikipedia.org/wiki/Principal_component_analysis**
[28]https://uk.mathworks.com/matlabcentral/fileexchange/24634-face-recognition-system-pca-based
[29]http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-9-135-139.pdf
[30]http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.4428
[31]https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2
[32]http://users.rsise.anu.edu.au/~wanglei/SPR_course/Direct_LDA.pdf