SEMINAR REPORT

Entitled

" FACE RECOGNITION USING OPENCV "

Submitted to the Department of Electronics Engineering
In Partial Fulfillment of the Requirement for the Degree of

Bachelor of Technology

(ELECTRONICS & COMMUNICATION)

Presented & Submitted By:

Mr. ANURAG AJEET KHARWAR

(Roll No. U19EC061)

B. TECH. III (EC), 5th Semester

Guided By:

Prof. A. H. LALLUWADIA

Associate Professor, ECED.

(Year : 2021-22)

DEPARTMENT OF ELECTRONICS ENGINEERING

Sardar Vallabhbhai National Institute of Technology

Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute of Technology

Surat-395 007, Gujarat, INDIA.

ELECTRONICS ENGINEERING DEPARTMENT

CERTIFICATE

This is to certify that the **SEMINAR REPORT** entitled **“Face Recognition using OpenCV”** is presented & submitted by Candidate **Mr. Anurag Ajeet Kharwar**, bearing **Roll No.U19EC061**, **of B.Tech. III, 5th Semester** in the partial fulfillment of the requirement for the award of **B. Tech.** degree in **Electronics & Communication Engineering** for academic year 2021-22.

He has successfully and satisfactorily completed his/her **Seminar Exam** in all respect. We, certify that the work is comprehensive, complete and fit for evaluation.

A. H. Lalluwadia

Prof. A. H. Lalluwadia
Associate Professor &
Seminar Guide

SEMINAR EXAMINERS :

Name of Examiner	**Signature with date**
1.Prof. M. C. Patel________	___________________
2.Dr. K. P. Upla_________	___________________

Dr. P. N. Patel
Associate Professor &
Head, ECED, SVNIT.

DEPARTMENT SEAL

(October - 2021)

Acknowledgement

I would like to thank respected Director of SVNIT Dr. SR. GANDHI, respected seminar guide Prof. A.H Lalluwadia and all the faculties' members for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

I also acknowledge with a deep sense of veneration, my gratitude towards my parents and members of my family who have always inspired and supported me morally.

I would also like to thank my friends who directly or indirectly helped me to complete this Seminar Report.

I further thank my Head of the Department Dr. Piyush N. Patel for his continuous and outstanding support in fulfilment of this task by providing me all related infrastructure and other facilities at the department for Seminar work.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs. Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free. Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

Anurag Kharwar
U19EC061
Sardar Vallabhbhai National Institute of Technology
Surat
27/10/2021

Abstract

Face Recognition using OpenCV

Face Recognition is a computer application that is capable of detecting, tracking, identifying or verifying human faces from an image or video captured using a digital camera. Face recognition in a real-time setting has an exciting area and a rapidly growing challenge Although lot of progress has been made in domain of face detection and recognition for security, identification and attendance purpose, but still there are issues hindering the progress to reach or surpass human level accuracy. These issues are variations in human facial appearance such as; varying lighting condition, noise in face images, scale, pose etc. This research paper presents a new method using Local Binary Pattern (LBP) algorithm combined with advanced image processing techniques such as Contrast Adjustment, Bilateral Filter, Histogram Equalization and Image Blending to address some of the issues hampering face recognition accuracy so as to improve the LBP codes, thus improve the accuracy of the overall face recognition system. I build a camera-based real-time face recognition system and set an algorithm by developing programming on OpenCV, Har Cascade, Eigenface, Fisher Face, LBPH, and Python. Our experiment results show that our method is very accurate, reliable and robust for face recognition system that can be practically implemented in real-life environment as an automatic attendance management system.

Signature of Student
Student Name : Anurag kharwar
Roll No. U19EC061
Guide Name :prof. A.H Lalluwadia
Seminar Date & Time : 27th Oct, 2021,4-6 pm

Examiners' Names :
Prof. M. C. Patel____________
Dr. K. P. Upla _____________

INDEX

Table of Contents

List of Figures

List of tables

Chapter 1
Introduction

When credit card or ATM are lost or stolen, any unauthorized can often come up with the correct pins (personal ID number). This growth in electronic revolutions has resulted in larger demand for fast and accurate user identification and authentication. The major problem is that the user of pins who is accessing is not verified.

To concur this problem and what can be betterment of the technology we will be discussing further in this chapter.

1.1 Overview

We are in the era of the information age where it is quickly getting revolutionized the way modern transaction are completed. Electronics being the important part of our everyday action, instead of pen and paper tradition. This growth in electronic revolutions has resulted in larger demand for fast and accurate user identification and authentication.

Codes, pins, passwords has vast use in banks accounts, computer systems or any kind of security gateways the major problem is that the user of pins who is accessing is not verified. When credit card or ATM are lost or stolen, any unauthorized can often come up with the correct pins (personal identification number).

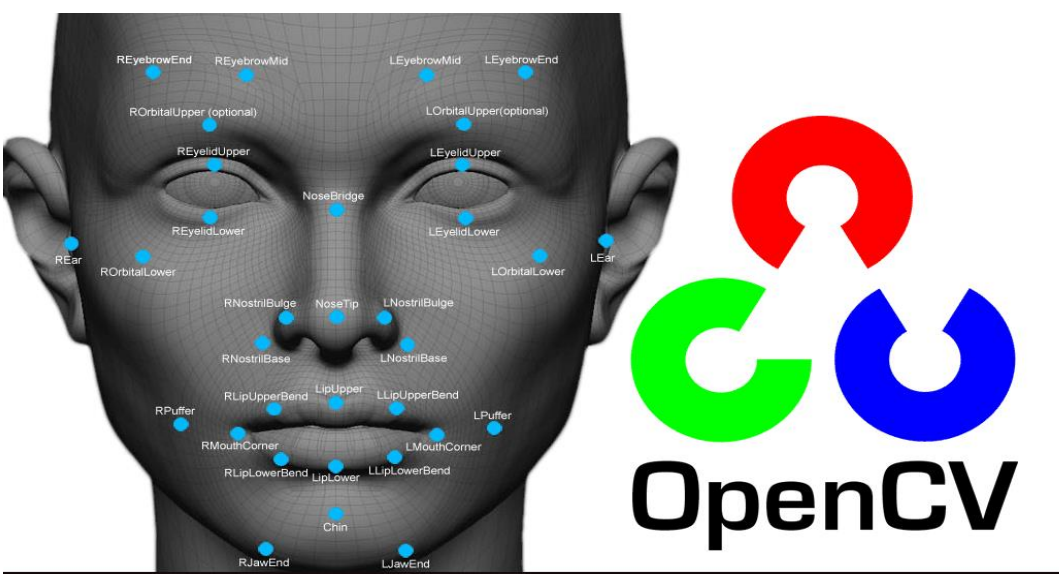


Figure 1.1: Basic parameter of a human face[1]

Courtesy

Despite knowing all the threat and warning, many individuals continue to create easily guessed PINs and passwords such that birthdays, phone numbers and even their own name or bank numbers. Recent cases of many frauds and theft have highlight that it requires for methods to prove that the person is truly who he/she claims to be.

Here we introduce you the Face recognition technology may solve this problem since a face has undeniably unique identification characteristic connected to its owner only expect in the case of identical twins. Its nontransferable. The system can then compare scans to records stored in a central or local database or even on a smart card, credit cards or any kind of public accessible cards.

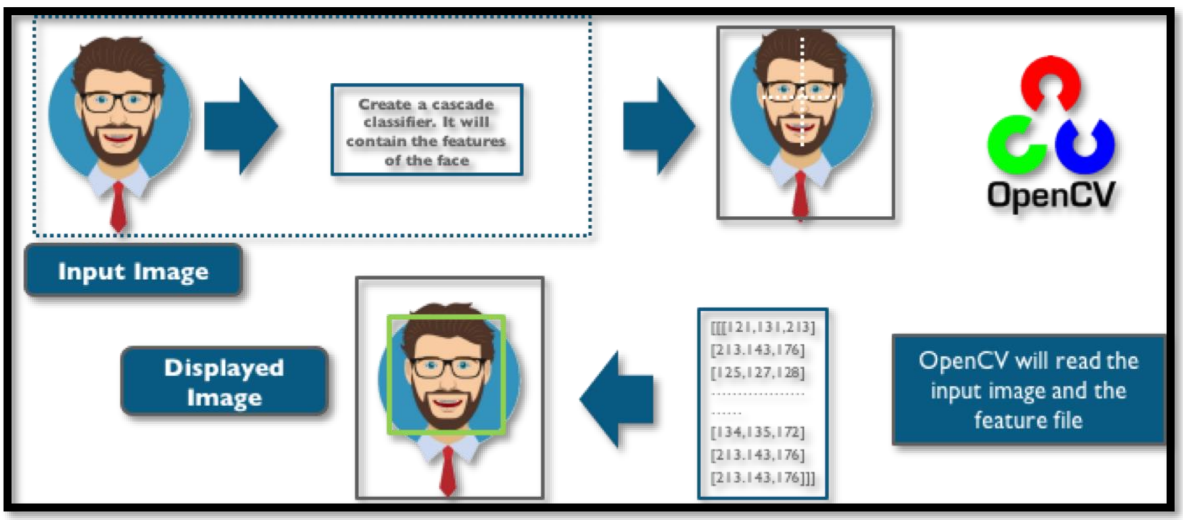


Figure 1.2: shows the face recognition Algorithm[2]
Courtesy

1.2 History of Face recognition

Face -recognition has very vast and old history as computer vision, both because of practical importance and need of the topic and interest of theoretical from cognitive scientist. Despite the fact that other methods of identification (such as fingerprints, or iris scans) can be more accurate, face recognition has always remained a major focus of research because of its non-invasive nature and because it is people's primary method of person identification.

Perhaps the most famous early example of a face recognition system is due to Korhonen , who demonstrated that a simple neural net could perform face recognition for aligned and normalized face images. The type of network has employed computed a face description by approximating the eigenvectors of the face images' autocorrelation matrix; these eigenvectors are now known as `eigenfaces.'

Kohonen's system was not a practical success, however, because of the need for precise alignment and normalization. In following years many researchers tried face recognition schemes based on edges, inter-feature distances, and other neural net approaches. While several were successful on small databases of aligned images, none successfully addressed the more realistic problem of large databases where the location and scale of the face is unknown.

Kirby and Sirovich (1989) later introduced an algebraic manipulation which made it easy to directly calculate the eigenfaces, and showed that fewer than 100 were required to accurately code carefully aligned and normalized face images. Turk and Pentland (1991) then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery, and to determine the precise location and scale of faces in an image.

They then demonstrated that by coupling this method for detecting and localizing faces with the eigenface recognition method, one could achieve reliable, real-time recognition of faces in a minimally constrained environment. This demonstration that simple, real-time pattern recognition techniques could be combined to create a useful system sparked an explosion of interest in the topic of face recognition.

1.3 Motivation

The most useful area in which face recognition is important is the biometrics that is used for authentication process which makes the work mor easier. Face recognition has very vast use in technology systems in which It has potential to perform tasks such as to have records generated in dataset in many aera such as offices, schools, colleges attendance system, it can be useful for the live surveillance system and helpful in catching thieves or the terrorist, can be helpful in the security of the common people and much required security aeras in the worldwide.

Government can also use Face-Recognition technology in order to verify the voters list, find missing person or child, find the no of peoples present at any public places, immigration process, also provide security over internet scams protecting Ecommerce and highly useful in medicine and healthcare range. This brings in a very demand or a real time face-recognition system, for several uses for the people and government.

Providing such excellent systems there would be ease in several activities these all various expects find me fascinating and motivates me to do future research on the facial recognition process

1.4 Applications

Face recognition system are useful in various expects such as security system, biometric scan and voting purpose, and law enforcement or justice solution by always being one step ahead of the world's ever advanced criminals.

This may include railways security, Airport security, computerized arrest and booking system on ticketing place and child base protection which is a software solution for global law enforcement agencies to help protect and recover missing sexually exploited children, particularly as it related to child pornography. It is also very useful in homeland defense which may include everything such as terrorist tempering or boarding aircraft, any infrastructure, popular hotels or any kind of terrorist activities can be stopped using the identification of known terrorists or similar kind of look over.

It is also applicable in any kind of terminal security. Face Recognition software, can improve the improve the security of the financial services industry, saving the institution time and money both through a reduction of fraud case and the administration expenses of dealing with forgotten password, furthermore biometric-based access control units can safeguard vaults, teller ensure that confidential information remains confidential while deterring identity theft, Particularly in the ATM system and card less transaction present on ecommerce transaction. It allows capturing, archiving, and retrieving identification characteristics such as any scar, tattoos, marks. We can also analysis scenes from either streaming or archived videos and looking forward for the ordinary occurrences, the detection of certain vehicles, Specific faces, etc.

This can save significant amount of time and money to those who spend long hours or days or even weeks monitoring video streams (I . e terrorist at Airport or any busy terminals, examining a bank security in criminal investigation.)

Figure 1.4(a) : Example of recognizing faces on the busy street.[3]
Courtesy

Face Recognition for Criminal Identification is a face recognition system in which the **security** expert will input an image of the person in question inside the system and the system will first preprocess the image which will cause unwanted elements such as noise to be removed from the image shown in below figure 2.2.

Figure 1.4(b):Face recognition technology used in terrorist or theft identification[4]
Courtesy

Figure 1.4(c) : Different age group detection technology based on face recognition.[5]
Courtesy

Our system simplifies the tedious tasks involved with monitoring employee time and attendance, employee tracking, and data collection. Our entirely automated time-tracking software "Minop" working in tandem with our biometric fingerprint, IRIS and face recognition devices — helps to improve workforce productivity and engagement. Our biometric time attendance system has capacity to support the requirements of various industries irrespective of size and complexity, from small ones to the very large ones as shown below figure 1.4(d)

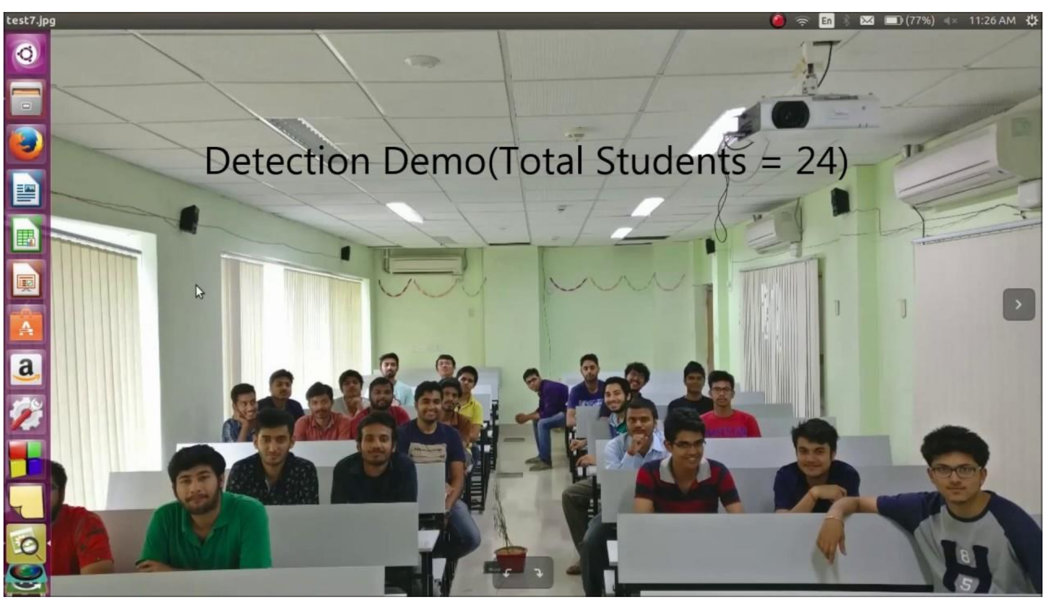


Figure 1.4(d) : Live Attendance based face recognition. [6]Courtesy

1.5 Report Outline

In this report

Chapter 1- covers the background and motivation for Face Recognition system using Open CV.

Chapter 2- Consists of summary of Literature survey of various researches.

Chapter 3 - Has covered various approach used in face recognition.

Chapter 4- Consists of brief description of the Face Recognition technology for their advantages and performance comparison between all approach.

Chapter 5- Consist of various use in security and biometric scan.

Chapter 6- Has covered the future recommendations in the Face Recognition.

Chapter 2
Literature Survey

FACE recognition is an important research problem spanning numerous fields and disciplines. This because face recognition, in additional to having numerous practical applications such as bankcard identification, access control, Mug shots searching, security monitoring, and surveillance system, is a fundamental human behaviour that is essential for effective communications and interactions among people. A formal method of classifying faces was first proposed in [1].

The author proposed collecting facial profiles as curves, finding their norm, and then classifying other profiles by their deviations from the norm. This classification is multi-modal, i.e. resulting in a vector of independent measures that could be compared with other vectors in a database. Progress has advanced to the point that face recognition systems are being demonstrated in real-world settings [2].

The rapid development of face recognition is due to a combination of factors: active development of algorithms, the availability of a large databases of facial images, and a method for evaluating the performance of face recognition algorithms. In the literatures, face recognition problem can be formulated as: given static (still) or video images of a scene, identify or verify one or more persons in the scene by comparing with faces stored in a database.

When comparing person verification to face recognition, there are several aspects which differ. First, a client – an authorized user of a personal identification system – is assumed to be co-operative and makes an identity claim. Computationally this means that it is not necessary to consult the complete set of database images (denoted model images below) in order to verify a claim. An incoming image (referred to as a probe image) is thus compared to a small number of model images of the person whose identity is claimed and not, as in the recognition scenario, with every image (or some descriptor of an image) in a potentially large database.

Second, an automatic authentication system must operate in near-real time to be acceptable to users. Finally, in recognition experiments, only images of people from the training database are presented to the system, whereas the case of an imposter (most likely a previously unseen person) is of utmost importance for authentication. Face recognition is a biometric approach that employs automated methods to verify or recognize the identity of a living person based on his/her physiological characteristics. In general, a biometric identification system makes use of either physiological characteristics (such as a fingerprint, iris pattern, or face) or behaviour patterns (such as hand-writing, voice, or key-stroke pattern) to identify a person. Because of human inherent protectiveness of his/her eyes, some people are reluctant to use eye identification systems.

Face recognition has the benefit of being a passive, non intrusive system to verify personal identity in a "natural" and friendly way. In general, biometric devices can be explained with a threestep procedure (1) a sensor takes an observation. The type of sensor and its observation depend on the type of biometric devices used. This observation gives us a "Biometric Signature" of the individual. (2) a computer algorithm "normalizes" the biometric signature so that it is in the same format (size, resolution, view, etc.) as the signatures on the system's database. The normalization of the biometric signature gives us a "Normalized Signature" of the individual. (3) a matcher compares the normalized signature with the set (or sub-set) of normalized signatures on the system's database and provides a "similarity score" that compares the individual's normalized signature with each signature in the database set (or sub-set).

What is then done with the similarity scores depends on the biometric system's application? Face recognition starts with the detection of face patterns in sometimes cluttered scenes, proceeds by normalizing the face images to account for geometrical and illumination changes, possibly using information about the location and appearance of facial landmarks, identifies the faces using appropriate classification algorithms, and post processes the results using model-based schemes and logistic feedback [3]. The application of face recognition technique can be categorized into two main parts: law enforcement application and commercial application. Face recognition technology is In this section we are basically discussed overview of the major techniques used in the face recognition system that apply mostly to the front frontal face.

primarily used in law enforcement applications, especially Mug shot albums (static matching) and video surveillance (real-time matching by video image sequences). The commercial applications range from static matching of photographs on credit cards, ATM cards, passports, driver's licenses, and photo ID to real-time matching with still images or video image sequences for access control. Each application presents different constraints in terms of processing. All face recognition algorithms consistent of two major parts: (1) face detection and normalization and (2) face identification. Algorithms that consist of both parts are referred to as fully automatic algorithms and those that consist of only the second part are called partially automatic algorithms. Partially automatic algorithms are given a facial image and the coordinates of the center of the eyes. Fully automatic algorithms are only given facial images.

On the other hand, the development of face recognition over the past years allows an organization into three types of recognition algorithms, namely frontal, profile, and viewtolerant recognition, depending on the kind of images and the recognition algorithms. While frontal recognition certainly is the classical approach, view-tolerant algorithms usually perform recognition in a more sophisticated fashion by taking into consideration some of the underlying physics, geometry, and statistics. Profile schemes as stand-alone systems have a rather marginal significance for identification, (for more detail see [4]). However, they are very practical either for fast coarse pre-searches of large face database to reduce the computational load for a subsequent sophisticated algorithm, or as part of a hybrid recognition scheme.

Such hybrid approaches have a special status among face recognition systems as they combine different recognition approaches in an either serial or parallel order to overcome the shortcoming of the individual components. Another way to categorize face recognition techniques is to consider whether they are based on models or exemplars. Models are used in [5] to compute the Quotient Image, and in [6] to derive their Active Appearance Model. These models capture class information (the class face), and provide strong constraints when dealing with appearance variation. At the other extreme, exemplars may also be used for recognition. The ARENA method in [7] simply stores all training and matches each one against the task image.

As far we can tell, current methods that employ models do not use exemplars, and vice versa. This is because these two approaches are by no means mutually exclusive. Recently, [8] proposed a way of combining models and exemplars for face recognition. In which, models are used to synthesize additional training images, which can then be used as exemplars in the learning stage of a face recognition system. Focusing on the aspect of pose invariance, face recognition approaches may be divided into two categories: (i) global approach and (ii) component-based approach. In global approach, a single feature vector that represents the whole face image is used as input to a classifier. Several classifiers have been proposed in the literature e.g. minimum distance classification in the eigenspace [9,10], Fisher's discriminant analysis [11], and neural networks [12]. Global techniques work well for classifying frontal views of faces. However, they are not robust against pose changes since global features are highly sensitive to translation and rotation of the face. To avoid this problem an alignment stage can be added before classifying the face. Aligning an input face image with a reference face image requires computing correspondence between the two face images.

The correspondence is usually determined for a small number of prominent points in the face like the center of the eye, the nostrils, or the corners of the mouth. Based on these correspondences, the input face image can be warped to a reference face image. In [13], an affine transformation is computed to perform the warping. Active shape models are used in [14] to align input faces with model faces. A semi-automatic alignment step in combination with support vector machines classification was proposed in [15]. An alternative to the global approach is to classify local facial components. The main idea of component based recognition is to compensate for pose changes by allowing a flexible geometrical relation between the components in the classification stage. In [16], face recognition was performed by independently matching templates of three facial regions (eyes, nose and mouth).

The configuration of the components during classification was unconstrained since the system did not include a geometrical model of the face. A similar approach with an additional alignment stage was proposed in [17]. In [18], a geometrical model of a face was implemented by a 2D elastic graph. The recognition was based on wavelet coefficients that were computed on the nodes of the elastic graph. In [19], a window was shifted over the face image and the DCT coefficients computed within the window were fed into a 2D Hidden Markov Model. Face recognition research still face challenge in some specific domains such as pose and illumination changes. Although numerous methods have been proposed to solve such problems and have demonstrated significant promise, the difficulties still remain. For these reasons, the matching performance in current automatic face recognition is relatively poor compared to that achieved in fingerprint and iris matching, yet it may be the only available measuring tool for an application. Error rates of 2-25% are typical. It is effective if combined with other biometric measurements. Current systems work very well whenever the test image to be recognized is captured under conditions similar to those of the training images. However, they are not robust enough if there is variation between test and training images [20]. Changes in incident illumination, head pose, facial expression, hairstyle (include facial hair), cosmetics (including eyewear) and age, all confound the best systems today.

As a general rule, we may categorize approaches used to cope with variation in appearance into three kinds: invariant features, canonical forms, and variation- modeling. The first approach seeks to utilize features that are invariant to the changes being studied. For instance, the Quotient Image [5] is (by construction) invariant to illumination and may be used to recognize faces (assumed to be Lambertian) when lighting conditions change. The second approach attempts to “normalize” away the variation, either by clever image transformations or by synthesizing a new image (from the given test image) in some

The methods include neural networks, hidden Markov model, face matching done geometrically and template matching. Eigenface is one of the most widely used methods in face recognition and detection which are broadly called as the principal components in mathematical terms. To represent the different amount of variation in the faces Eigen vectors are used.

Face detection are highly based on Neural network. An artificial eural network(ANN) Was used in face recognitions which contained a single layer Which shows adaptiveness in crucial face recognition systems. The face verification is done using a double layer of WISARD in neural networks. Graph matching is other option for face recognition. The object as well as the face recognition can be formulated using graph matching performed by optimization of a matching function.

Hidden Markov Models is the way by which stochastic modeling of nonstationary vector time series based on HMM model applied to the human face recognition wherein the faces get divided into parts such as the eyes, nose, ears, etc. The face recognition and correct matching is 87% correct as it always gives out the best and right choice of face detection through stored dataset. Or else the relevant model reveals the identity of the face. The geometrical feature matching is the technique which is based on the geometrical shapes of the face.

The geometrical face configuration has sufficient dataset for face detection
and recognition system. This is one of the commonly used method of the face recognition and detection. This system apparently gives satisfactory results. Template matching is one of the techniques through which the test image is represented as a two- dimensional array of values which can be compared using Euclidean distance with single template representing the whole face.
This method can also use more than one face template from different points of view to represent an individual face.

Face -recognition has very vast and old history as computer vision, both because of practical importance and need of the topic and interest of theoretical from cognitive scientist. Despite the fact that other methods of identification (such as fingerprints, or iris scans) can be more accurate, face recognition has always remained a major focus of research because of its non-invasive nature and because it is people's primary method of person identification.

Perhaps the most famous early example of a face recognition system is due to Korhonen , who demonstrated that a simple neural net could perform face recognition for aligned and normalized face images. The type of network has employed computed a face description by approximating the eigenvectors of the face images' autocorrelation matrix; these eigenvectors are now known as `eigenfaces.'

Chapter 3
Face Recognition

Need for face recognition

- Greater demand for fast and accurate users' identification and authentication.
- Using the proper PIN gains and access, but the user of PIN is not verified.
- What if credit or ATM card is stolen and an unauthorized user somehow
  Come up with the correct personal code.
- It is one of many Biometric such as finger scan, Iris scan, Retina scan, Hand scan etc.
- Significance increases in Electronics Transaction

3.1 Face recognition over Biometric scan.

- It requires no physical interaction on behalf of the user.
- It is accurate and allows for high enrolment and verification rates.
- It does not require an expert to intercept the comparison result.
- It can use the existing hardware infrastructure, camera and image already
  Captured without causing ant problem to the device.
- It is the only biometric that allows passive identification in any environment (e.g., Identifying a terrorist in busy Airport)

hairstyles, and other details in your photos, the better the software's ability to pick you out in other photos. It's one thing to confine this technology to the photos on your own computer, but things get more complicated when the internet gets involved. Facebook got in trouble with privacy advocates when it rolled out facial recognition by default. It's since dialed it back to "Tag Suggestions," which you can choose to disable. Even if you disable it, though, Facebook still collects information about your face whenever it's tagged. And when you consider that Facebook's 600 million members upload over 250 million photos every day, you see that they're building an empire of facial data. Rumor has it they're building a way to search for people by picture alone. And Google's Goggles app can already identify inanimate objects through photographs. Add already-existing facial recognition software to that, and you could "identify strangers on the street."

3.2 Facial recognition use in today's life.

Facial recognition is still alive and well. It's utilized in a variety of settings, including social networking, picture editing, security, police enforcement, casinos, and unexpected locations. For instance, the dating service FindYourFaceMate.com uses facial recognition to compare user photographs, while DoggelGanger.com links potential dog owners with dogs that look like them.

Face recognition cameras scanned all the fans walking through the turnstiles at Super Bowl XXXV, now referred to as the Snooper Bowl, running the scans against a database of criminal mugshots. That was a decade ago, when the internet was still in its relative childhood.

We're in an age now when Facebook Instagram snapchat or many more collects 100-page dossiers on all of us, when ad networks track everything we do online, when companies buy and sell our contact information: the street we grew up on, the names of our family members, aerial shots of our homes. SceneTap's interface. Before going to a pub, you may use the app to verify the gender ratio. Facial recognition is also used in a lot of smartphone apps. Scene Tap, for example, keeps track of male-to-female ratios and ages at 250 participating bars across the United States. Face bars are installed using these bars. Face recognition cameras scanned all of the people entering the Snooper Bowl, now known as Super Bowl XXXV, and matched the scans to a database of criminal mugshots. That was almost a decade ago, when the internet was just getting started. We now live in a world where Facebook, Instagram, Snap Chat, and other social media companies create 100-page dossiers on each of us, ad networks watch everything we do on the internet, and The app, which has about 30,000 downloads, uses the same facial-recognition technology deployed by local law enforcement to identify criminal suspects, says Animetrics CEO Paul Schuepp. Companies give two main reasons for using facial recognition technology: it helps with security, and it makes photo editing and sharing easier. On the security side, law enforcement officials have argued that facial recognition can help find missing people, identify criminals in a crowd, preempt terrorists from boarding planes with fake passports.

It's also used for private security in casinos to identify card counters and kick them out before they can win too much. Casinos also say their systems identify people with gambling addictions who've asked casinos to forcibly remove them if they can't stop themselves. Even supermarket security use face recognition: in the United Kingdom, one supermarket chain utilizes facial recognition to prevent underage consumers from purchasing alcohol. Facial recognition in picture sharing may scan albums for faces and either recommend or automatically tag individuals. Apple's iPhoto, Google's Picasa web, Microsoft's Windows Live Photo Gallery, and other photo editors already include it. It also collects data on other people's faces using existing tags: the more tags, angles, and lighting kinds there are, the better.

3.3 Face detection and face recognition

- Face Verification
- Face or no face decision
  among all images
- Face Identification
- Identify who’s image

3.3.1 Type of face recognition

Based on local Regions

- Local Feature Analysis (LFA)
- Gabor Wavelet

Based on Global Appearance

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)

- The following sections describe the face recognition algorithms
  - Eigenface
  - Fisher face
  - Local binary pattern histogram and how they are implemented in OpenCV.

3.4 Eigenface

Eigenface is based on PCA, which uses a series of pictures to categories and extract features. It's crucial that the photos have the same lighting and that the eyes in each image match. In addition, all pictures utilized in this approach must be grayscale and have the same amount of pixels. Consider the picture in figure 4 with n × n pixels for this example. A 12n2 matrix is created by concatenating each raw into a vector. The pictures in the dataset are kept in a single matrix, with columns matching to the number of photographs. To achieve an average human face, the matrix is averaged (normalized). Unique characteristics to each image vector are created by removing the average face from each image vector.. A simplified illustration can be seen in figure 3.2.

3.4.1 Eigenface approach

- Eigenfaces are the Eigenvectors of Covariance matrix of dataset.
- Eigenfaces are also referred to as GHOSTLY images.
- Prime reason ->To represent the input data efficiently -each individual face Can be represented in term of linear combination of eigenfaces.
- Need to reduce dimensionality -------Principal Component Analysis (PCA)

3.4.2 Principal component analysis

- It lowers the dimension of data by removing useless information and properly decomposing the face structure into orthogonal main components, which we call 'Eigenfaces.'
- Face space forms a cluster – PCA gives suitable representation below Figure 3.1

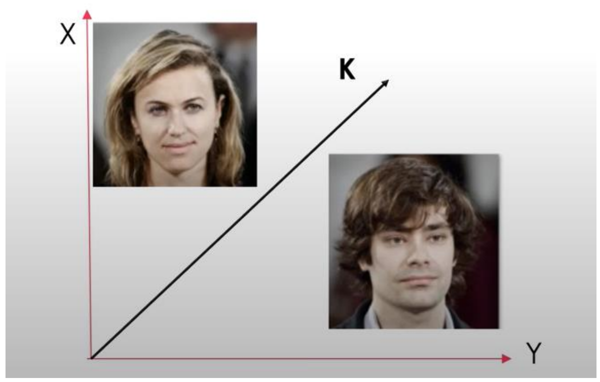


Figure 3.4.2(a) : Represent principal component analysis.[7]
Courtesy

RECOGNITION PROCESS

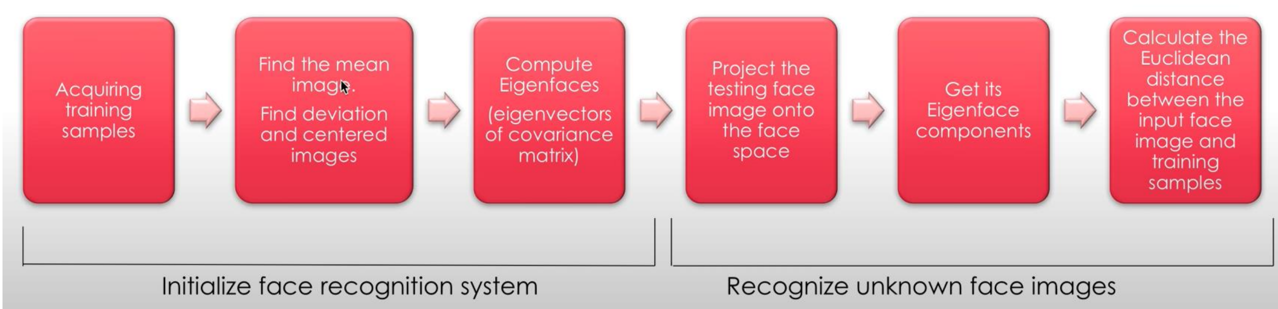


Figure 3.4.2(b) : Show algorithm of face recognition process[8]
Courtesy

FIRST STEP– TO OBTAIN DATA

Obtain a set of images which form a data set and will be considered for face Recognition Process.

SECOND STEP – TO OBTAIN THE MEAN IMAGE

Calculate the mean image

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_{i}$$

Figure 3.4.2(d)- Output after applying mean filter[9]

THIRD STEP-PROJECT THE FACE IMAGE INTO THE FACE SPACE

Find the deviation between input image and the mean image by: -

$$\Phi_i = \Gamma_i - \Psi$$

Then the centered image matrix is found based on this difference.

FOURTH STEP -COMPUTE ELIGENFACES

Covariance matrix is found using the centered image by the formula

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_{n}\Phi_{n}^{T}$$

The eigen value are then found for this co-variance matrix and hence eigenvectors are computed.

And eigenvectors that are less than a specific threshold value are eliminated by PCA

When the eigen vectors are projected on face a set of 'Ghostly' images i.e., eigenfaces are formed.

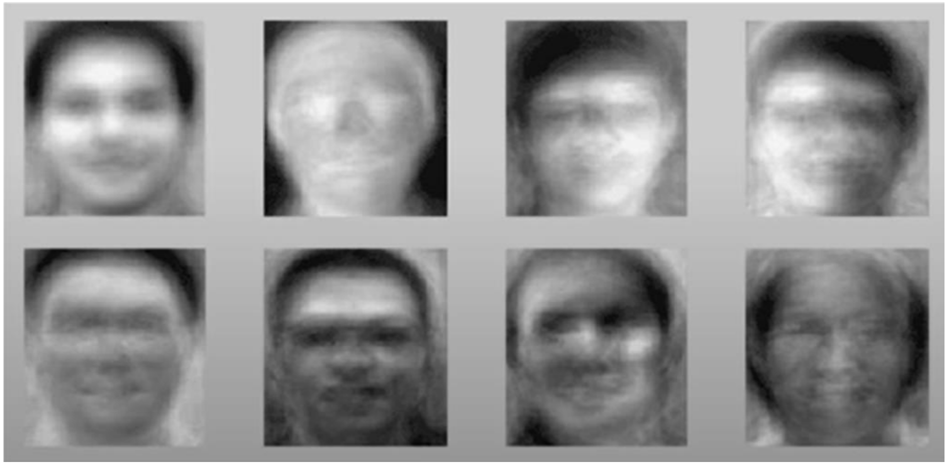


Figure 3.4.2(e): shows Eigenfaces obtained from above equation.[10]
Courtesy

FINAL STEP-RECOGNISE IMAGE

Image that needs to be recognized is fed into the program

By the previous four steps the eigen components of the image are obtained.

- COMPUTE EUCILDEAN DISTANCE

  The Euclidean distance between the eigenface of the test image and the previous computed eigenfaces are evaluated.

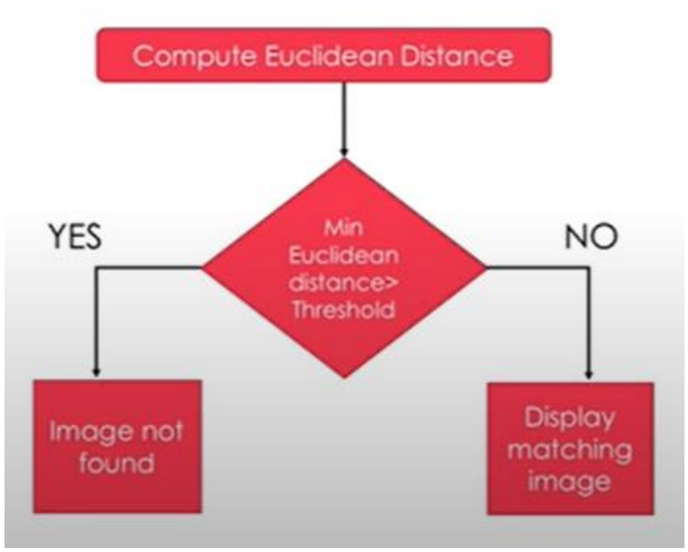


Figure 3.4.2(f) : Euclidean Distance between eigenface and previous computed eigenface.[11]
Courtesy

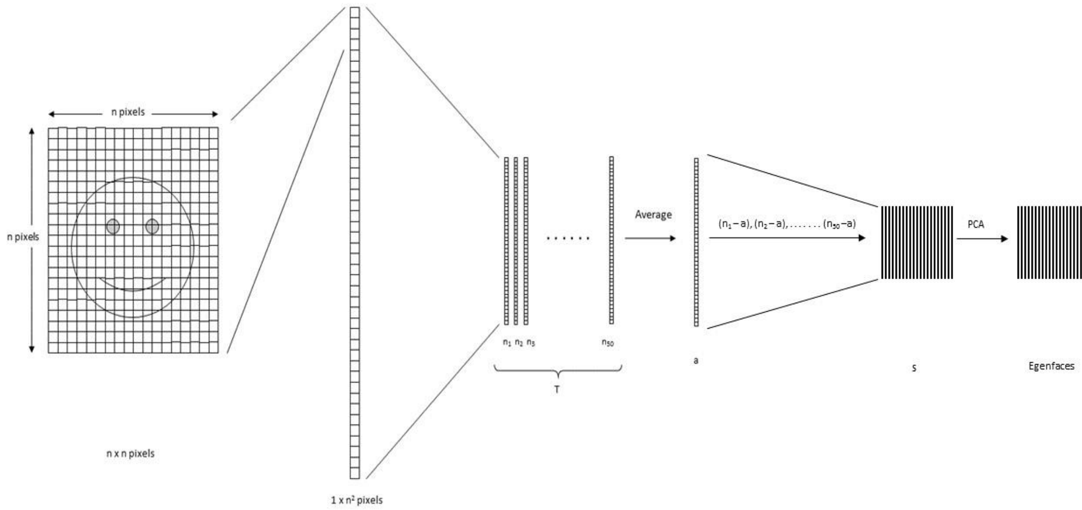


Figure 3.4.2(g): Pixels of the image are reordered to perform calculations for Eigenface[12]
Courtesy

Now the further step is to computing the covariance matrix from the obtained result. To get the Eigen vectors from the dataset, Eigen analysis is performed using the principal component analysis. From the result, where covariance matrix is the diagonal, where it has the highest variance is considered the first Eigen vector. Second Eigen vector is the direction of the next highest variance, and it is in 90 degrees to the first vector. Third will be the next highest variation, and so on. Each column is considered an image and visualized, resembles a face and called Eigenfaces. When a face is required to be recognized, the image is imported, resized to match the same dimensions of the test data as mentioned above. Further more weight can be calculated by projecting the features on to each of the Eigenfaces. These weights correspond to the similarity of the features extracted from the different image sets in the dataset to the features extracted from the input image. By comparing with the whole data set the input image can be identified as a face. In order to get to know by which person that face belong we can compare it to the subset. To eliminate and control false detection and recognition we can apply threshold detection and identification. PCA is sensitive to large numbers and assumes that the subspace is linear. If the same faces are analyzed under the dissimilar lighting conditions, it will mix the values when distribution is calculated and impossible to classified effectively. This makes to different lighting conditions poses a problem in matching the features as they can change dramatically.

3.5 Fisher Face

Fisher face technique basically based upon the Eigenface and is based on LDA (Linear discriminant analysis) derived from Ronald Fishers' linear discriminant technique which is used for pattern recognition. Data point information as well as classes uses labels. When reducing dimensions, Principal component analysis looks at the greatest variance, while LDA (Linear discriminant analysis), using labels, looks at an interesting dimension such that, when you project to that dimension you maximize the difference between the mean of the classes normalized by their variance. Linear discriminant analysis the ratio of the between-class scatter and within-class scatter matrices. Because of this, different lighting conditions in images has a limited effect on the classification process using LDA (Linear discriminant analysis) technique. Eigenface maximizes the variations while Fisher face maximizes the mean distance between and different classes and minimizes variation within classes. This enables LDA (Linear discriminant analysis) to differentiate between feature classes better than PCA and can be observed in figure. After all it takes less amount of space and is the fastest algorithm in this project. Because of these PCA (Principal component analysis) is more suitable for representation of a set of data while LDA is suitable for classification.

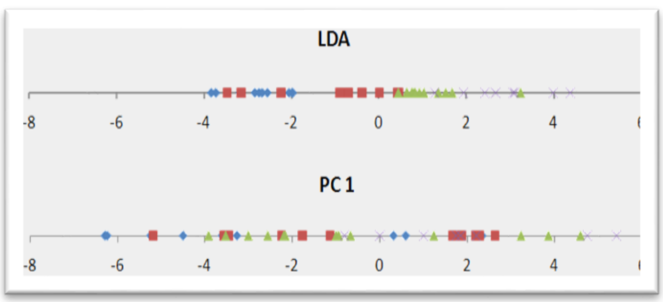


Figure 3.7: The first component of PCA and LDA. Classes in PCA looks more mixed than of LDA[13]

Fishermen implementation

Steps:

- Apply PCA to reduce dim., user fisher LDA multi-class independent variable (c-1 dimension).
- Construct the centered matrix, calculate average face (mean-image V)
- Calculate covariance matrix, find eigenvectors and values
- Project all training images onto subspace maximize class separation.

- Testing
    - Select a test image, project into subspace above
    - Measure distance between the new face and template of faces in training set.
    - Classes are distinguished by closest face training dataset.
    - Repeat for other test faces

Table 1. Comparison of Eigenface and Fisherface method

Criteria	Eigenface Approach	Fisherfacce Approach
Error rate dealing with variations	Higher than fisherface	Lower than Eigenface
Total scatter across all images	Maximum total scatter	Maximum between- class scatter
Implement PCA	Yes	Yes
Implement LDA	No	yes
Variation in lighting and pose	Almost always larger than image variation	Adaptable
Discriminating accuracies	Not always optimal	Better than eigenface
Between class and within class scatter	Scatter is maximized for both between- class and within-class	Scatter is maximized between-class
Class-specific method class selections	No	Yes
Images identification in the training k-dimensional subspace	Classify the image off the face by the closest face.	Distinguish classes by the closet face

3.6 Local Binary Pattern Histogram

In the year 1990 Local binary patterns were proposed as classifiers in computer vision. The combination of Local binary pattern with histogram-oriented gradients was introduced in the year 2009 that increased its performance in certain datasets. For the encoding, the image has to divided into cells (4 x 4 pixels). Using a clockwise or Anti-clockwise direction central pixel values are compared with the surrounding as shown in figure 6. The value of intensity or luminosity of each neighbor is compared with the center pixel. Depending if the difference is lower or higher than a 1, 0 or 0 is assigned to the location. The 8-bit value result will be provided to the cell. The major beneficial advantage of this technique is even if the luminosity of the image as shown below

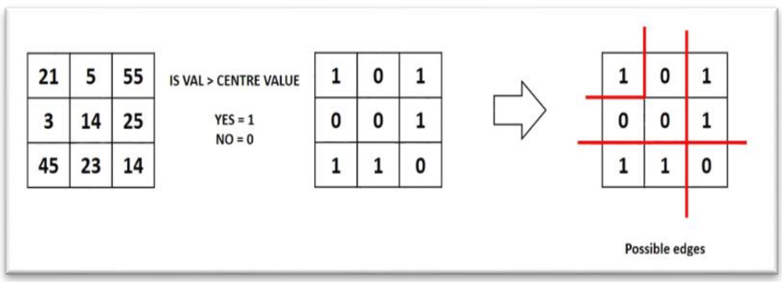


Figure 3.6(a): Local binary pattern histogram generating 8-bit number[14]

is changed as in figure 3.6(a), There is no change in the outcomes. to expedite the process In bigger cells, histograms are used to determine the frequency of value occurrences. Edges can be discovered by evaluating the results in the cell as the values change. We may construct feature vectors by calculating the values of all cells and condensing the histograms. Images having an ID can be categorized using processing. The same technique is used to classify the input pictures, which are then compared to the dataset to calculate the distance. We may acquire recognized and unknown faces by setting threshold values.. Eigenface and Fisher face compute the dominant features of the whole training set while LBPH (Local binary pattern histogram) analyze them individually.

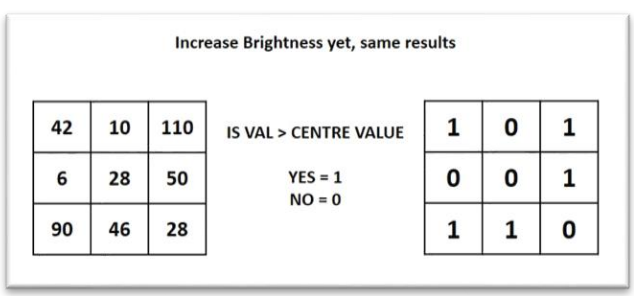


Figure 3.6(b): The results are same even if brightness is changed[15]

Chapter 4
Face Recognition Methodology

Hybrid Methods: Hybrid face recognition systems use a **combination of both holistic and feature extraction methods**. Generally 3D Images are used in hybrid methods. The image of a person's face is caught in 3D, allowing the system to note the curves of the eye sockets, for example, or the shapes of the chin or forehead.

4.1 Methodology

The technique and needed explanations of the apps used for face detection, data collection, face training, and face recognition are provided.. The entire project was coded in Python using a mixture of IDLE and PyCharm IDEs and using library used are Opencv2, OS, NumPy etc.

First initial stage was creating a face detection system using Haar-cascades algorithm. To create a new Harr-cascade training is required, OpenCV has a vigorous set of Haar-cascades that was used for the project. Using face-cascades alone may cause any random objects to be identified and using specified only eye cascades were incorporated to obtain stable face detection.

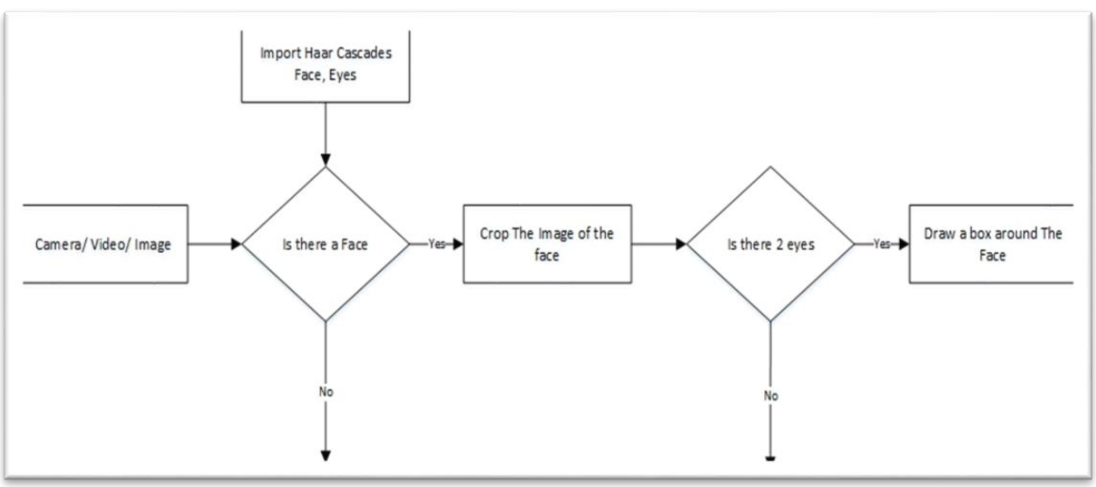


Figure 4.1 : The flowchart of the face detection system[16]

The flow chart of the face detection application classifier objects are produced in OpenCV using the classifier class through cv2.CascadeClassifier() and importing the corresponding XML files. To capture pictures, a camera object is constructed using cv2.VideoCapture(). By making use of the CascadeClassifier.

Objects of varied sizes are matched by detectMultiScale(), and the location is returned. The face is trimmed using the location data for additional verification. Eye cascade is used to confirm that the chopped face has two eyes. If the condition is met, a marking is put around the face to show that a face was spotted in the location Figure 4.1 .

4.2 Face Recognition Process

For this project three algorithms are implemented independently. These are Eigenface, Fisherface and Linear binary pattern histograms respectively. All three can be implemented using OpenCV libraries. There are three stages for the face recognition as follows:

1. Collecting images IDs
2. Extracting unique features, classifying them and storing in XML files
3. Matching features of an input image to the features in the saved XML files and predict identity

4.2.1 Collecting The Image Data

Typically, categorization pictures are collected manually using photo editing tools to crop and resize photos. Furthermore, for proper functioning, PCA and LDA require the same amount of pixels in all pictures. This time-consuming and labor-intensive job is mechanized by an application that collects 50 pictures with various emotions. The application detects suitable expressions between 300ms, straightens any existing tilt and save them. The Flow chart for the application is shown in figure 4.2.

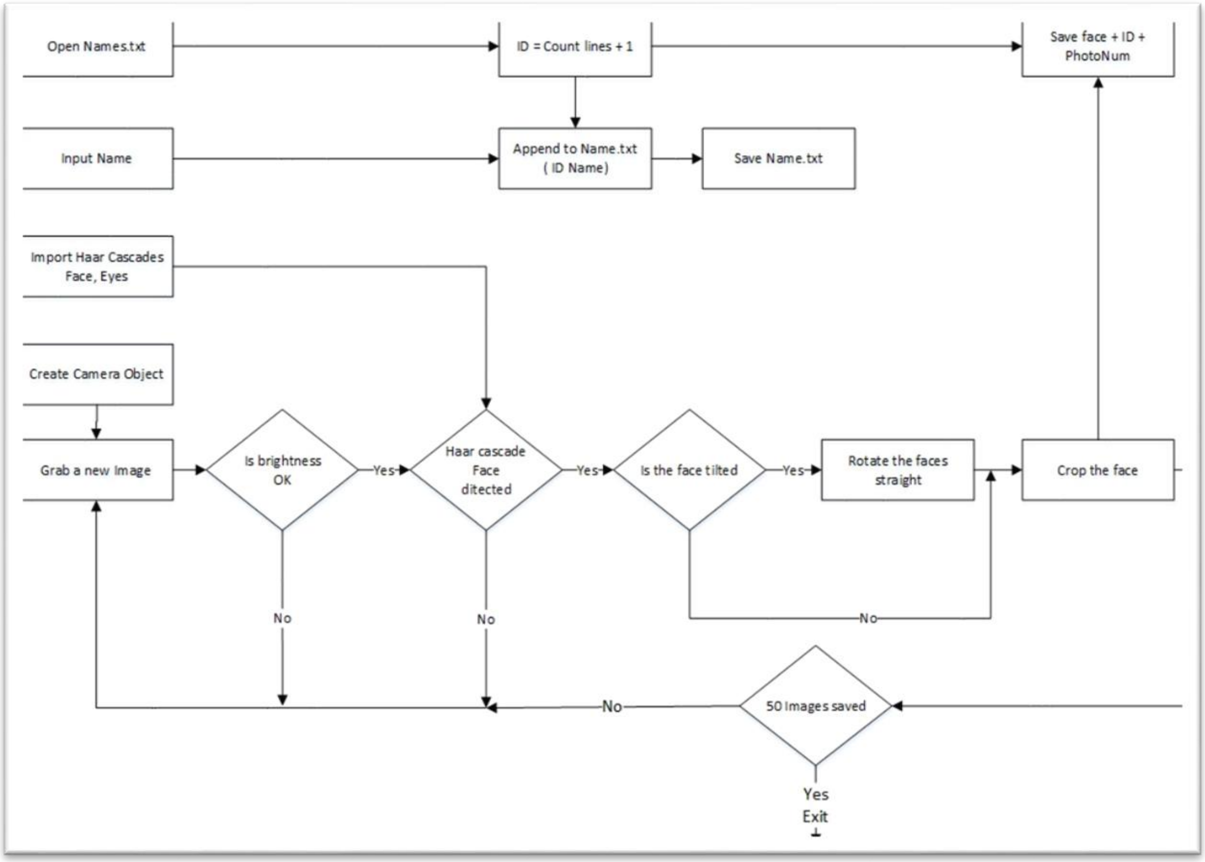


figure 4.2.1: The Flowchart for the image collection[17]

The application begins with a request for a name to be typed and saved in a text file with the ID. The first part is initiated by the facial detection system. However, before beginning the capture, the programmed checks the brightness levels and will only capture if the face is properly lighted. Following the detection of the face, the location of the eyes is examined. The programmer automatically corrects the orientation whenever the head is tilted. These two additions were built with the Eigenface algorithm in mind. The image is then cropped and stored with the ID as the filename so that it may be identified later. This software is looped until 50 valid pictures are obtained from the individual..This application made data collection efficient.

- **Codes for collecting the samples**

```
import cv2 as cv

face_classifier =
cv.CascadeClassifier('C:/Users/Anura/PycharmProjects/pythonProject5/FaceDetection-
HaarCascade-master/FaceDetection-HaarCascade-
master/haarcascade_frontalface_default.xml')


def face_extarctor(img): // cre

    gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,5)

    if faces is():
        return None

    for(x,y,w,h) in faces:
        cropped_face = img[y:y+h, x:x+w]

    return cropped_face

cap = cv.VideoCapture(0)

count=0
while True:
    ret, frame = cap.read()

    if face_extarctor(frame) is not None:
        count += 1
        face = cv.resize(face_extarctor(frame),(200,200))
        face = cv.cvtColor(face, cv.COLOR_BGR2GRAY)
        cv.putText(frame, 'face found', (50, 50), cv.FONT_ITALIC, 1, (255, 255, 0), 2)

       file_name_path = ('C:/Users/Anura/PycharmProjects/pythonProject5/faces/INPUT' +
str(count) + '.jpg')
        cv.imwrite(file_name_path, face)
        cv.putText(face, str(count), (50, 50), cv.FONT_ITALIC, 1, (255, 255, 0), 2)
        cv.imshow('face cropper', face)
    else:
```

```
        print("face not found")

        pass

    if cv.waitKey(1) == 13 or count == 50:
        break

cap.release()
cv.destroyAllWindows()
print("collecting sample complete")
```

4.2.2 Training the classifiers

The Face Recognizer class in OpenCV allows the generation of XML files to hold features collected from datasets. Imported photos are converted to grayscale and saved with IDs in two lists with the same indexes. The face recognizer class is used to build Face Recognizer objects. Each recognizer can accept the following parameters:

cv2.face. createEigenFaceRecognizer ()

1. Takes in the number of components for the PCA for crating Eigenfaces. OpenCV documentation mentions 80 can provide satisfactory reconstruction capabilities.
2. Takes in the threshold in recognizing faces. If the distance to the likeliest Eigenface is above this threshold, the function will return a -1, that can be used state the face is un recognizable **cv2.face. createFisherfaceRecognizer ()**
3. The Face Recognizer class in OpenCV allows the generation of XML files to hold features collected from datasets. Imported photos are converted to grayscale and saved with IDs in two lists with the same indexes. The face recognizer class is used to build Face Recognizer objects. Each recognizer can accept the following parameters:
4. The first argument is the number of components for the LDA for the creation of Fisherfaces.OpenCV mentions it to be kept 0 if uncertain.
5. Similar to Eigenface threshold. -1 if the threshold is passed.

cv2.face.createLBPHFaceRecognizer()

1. The radius from the center pixel to build the local binary pattern.
2. The Number of sample points to build the pattern. Having a considerable number will slow down the computer.
3. The Number of Cells to be created in X axis.
4. The number of cells to be created in Y axis.
5. A threshold value similar to Eigenface and Fisher face. if the threshold is passed the object will return -1

Images are imported, resized, transformed into NumPy arrays, and saved in a vector using recognizer objects. The picture ID is obtained by dividing the file name and saved in another vector.. By using **FaceRecognizer.train(NumpyImage, ID)** all three of the objects are trained. It must be noted that resizing the images were required only for Eigenface and Fisher face, not for LBPH. Next, the configuration model is saved as an XML file using **FaceRecognizer.save(FileName)**. In this project, all three are trained and saved through one application for convenience. The flow chart for the trainer is shown in figure 4.4.

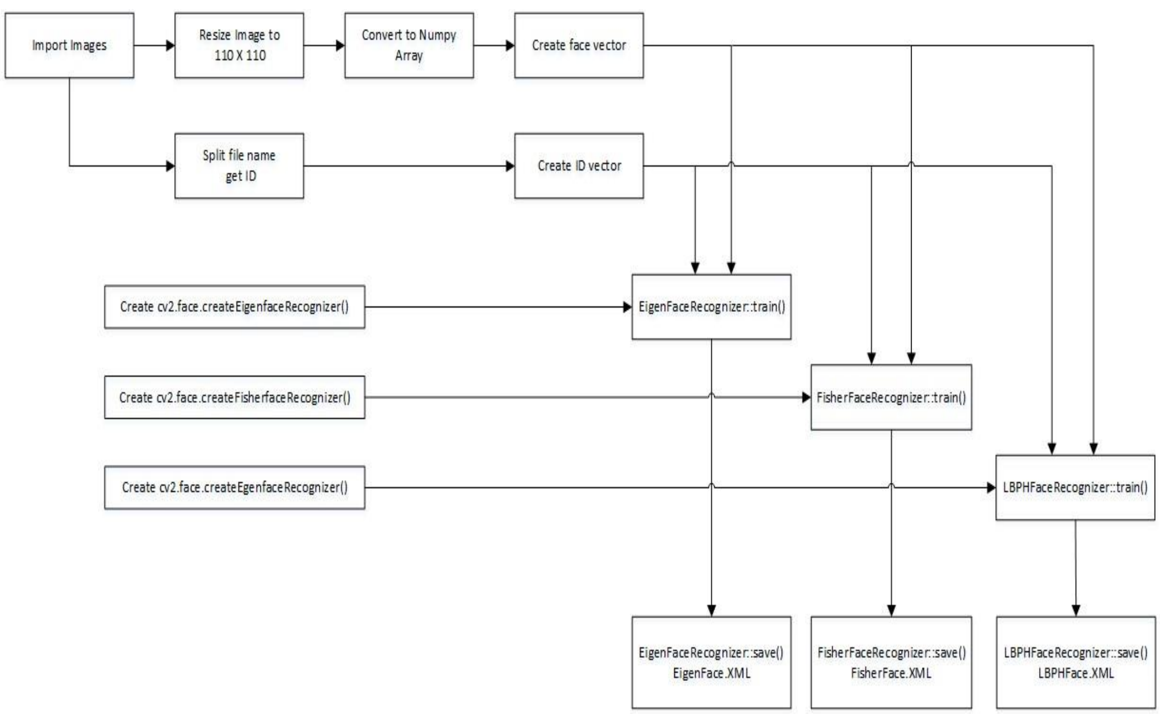
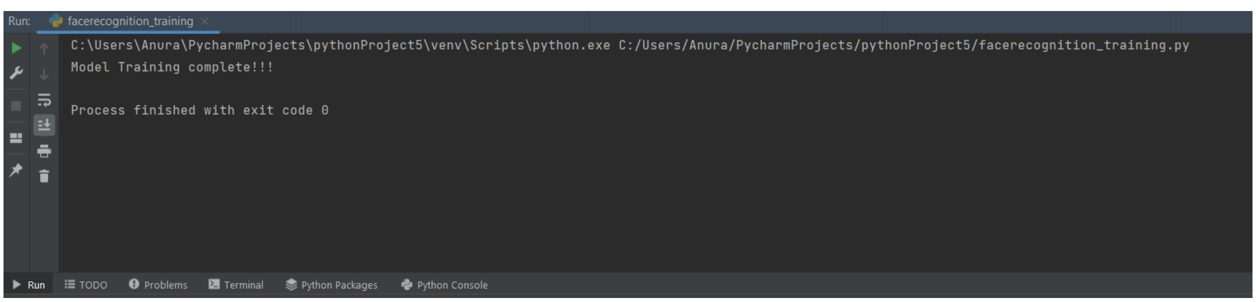


Figure 4.2.2: Flowchart of the training application[18]
Courtesy

- **Codes for training the classifier**

```
import NumPy as np
import cv2 as cv

from os import listdir
from os.path import isfile, join

data_path = 'C:/Users/Anura/PycharmProjects/pythonProject5/faces/'
onlyfiles = [f for f in listdir(data_path) if isfile(join(data_path, f))]

Training_data, Labels = [], []

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

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

model = cv.face.LBPHFaceRecognizer_create()


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

print("Model Training complete!!! ")
```

output terminal ..



4.2.3 The Face Recognition

The desired parameters are used to construct the face recognizer object. Face detectors are used to identify faces in images that have been cropped and transferred so that they may be identified. This is accomplished using the same method as the picture capturing program.. For each face detected, a prediction is made using **FaceRecognizer.predict()** which return the ID of the class and confidence. The process is same for all algorithms and if the confidence his higher than the set threshold, ID is -1. Finally, names from the text file with IDs are used to display the name and confidence on the screen. If the ID is -1, the application will print unknown face without the confidence level. The flow chart for the application is shown in figure 4.2.3.

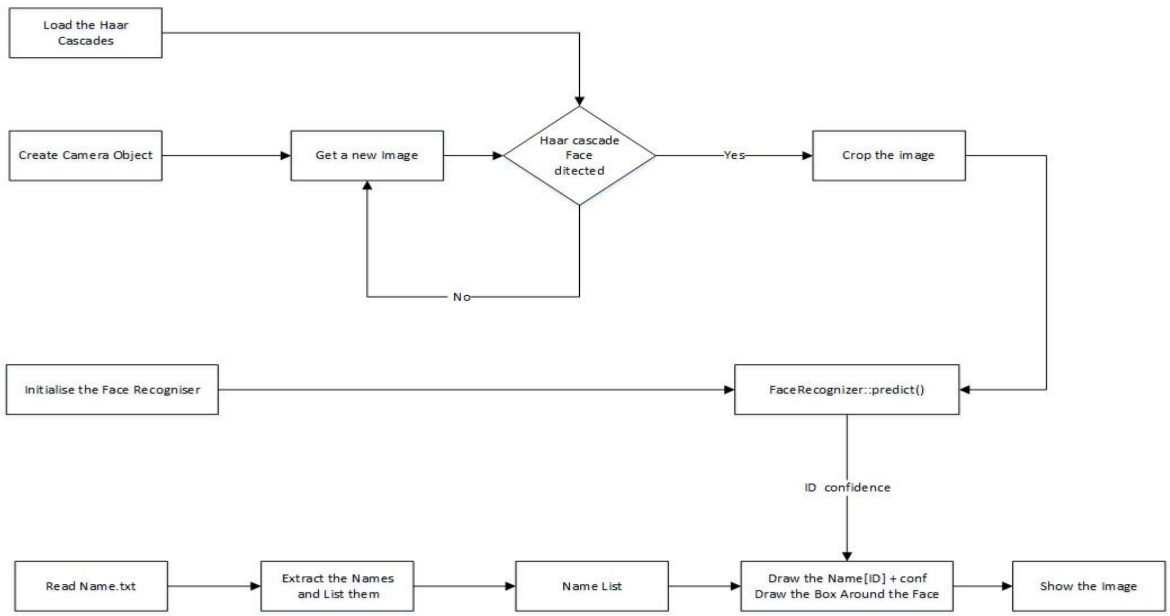


Figure 4.2.3 : Flowchart of the face recognition application[19]

- **Final Code For Face Recognition**

```
import cv2
import numpy as np
import cv2 as cv

from os import listdir
from os.path import isfile, join

data_path = 'C:/Users/Anura/PycharmProjects/pythonProject5/faces/'
onlyfiles = [f for f in listdir(data_path) if isfile(join(data_path, f))]

Training_data, Labels = [], []

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

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

model = cv.face.LBPHFaceRecognizer_create()


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

print("Model Training complete!!! ")

face_classifier = cv.CascadeClassifier('C:/Users/Anura/PycharmProjects/pythonProject5/FaceDetection-
HaarCascade-master/FaceDetection-HaarCascade-master/haarcascade_frontalface_default.xml')


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

    if faces is ():
        return img, []

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

    return img, roi


cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
```

```
    image, face = face_detector(frame)

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

        if result[1] < 500:
            confidence = int(100*(1-(result[1])/400))
          #  display_string = str(confidence)+'% confidence it is user'
        #cv.putText(image, display_string, (100, 120), cv.FONT_HERSHEY_SIMPLEX, 1, (250, 120, 240), 2)

        if confidence > 85:
            cv.putText(image, "unlocked", (250, 450), cv.FONT_HERSHEY_SIMPLEX, 1, (0, 225, 0), 2)
            display_string = str(confidence) + '% confidence it is user'
            cv.putText(image, display_string, (100, 120), cv.FONT_HERSHEY_SIMPLEX, 1, (250, 120, 240), 2)
            cv.imshow('face_cropper', image)
            print("USER Face detected....")
        else:
            cv.putText(image, "locked ", (250, 450), cv.FONT_ITALIC, 1, (0, 0, 255))
            display_string = str(confidence) + '% confidence it is NOT user'
            cv.putText(image, display_string, (100, 120), cv.FONT_HERSHEY_SIMPLEX, 1, (250, 120, 240), 2)
            cv.imshow('face_cropper', image)


    except:
        cv.putText(image, "No face found", (250, 450), cv.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255))
        cv.imshow('face_cropper', image)
        print("No face found!!!!!!!")
        pass

    if cv.waitKey(1) == 13:
        break

cap.release()
cv.destroyAllWindows()
```

Chapter 5
Implementaionn and model testing

In this chapter we will be discussing about the various technique used in this report and result e obtained from the dataset along with the accuracy as earlier we discussed we implemented our project using LBPH technique so we will see the output we obtained.

5.1 Testing of model and output

The photos that were gathered are presented below. Each face contains 50 pictures. To run over the parameters of each algorithm, three apps were created. The algorithm is trained using multiple parameters and evaluated against a snapshot on each iteration. Following the completion of the testing, the resultant data is shown.. The applications are :

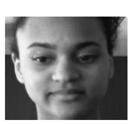
TestDataCollector EigenFace.py

TestDataCollector FisherFace.py

TestDataCollector LBPH.py.

ID1

ID: 2

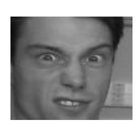


ID: 3

ID: 13

ID: 14

ID: 16



ID: 17

The first test image is shown in figure and the plots are analyzed below.

The resulting ID change is plotted below in figure 6.1. Note when components were 1, it identified

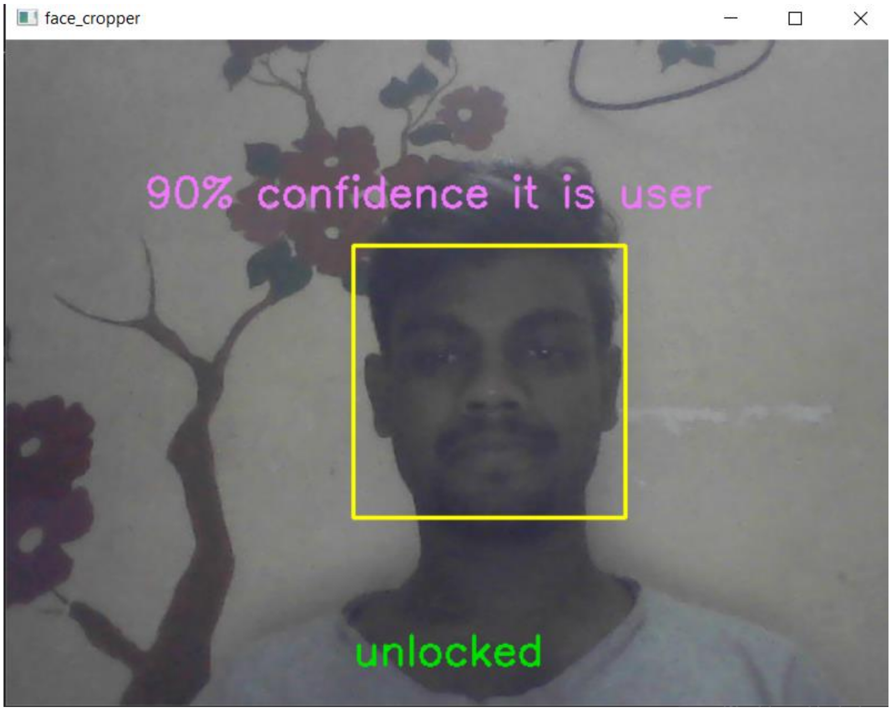


Figure 5.1(a): Testing and training test image[20]

the face as ID-17 and the rest are between ID-20 and ID-21, which is the same person. The change of

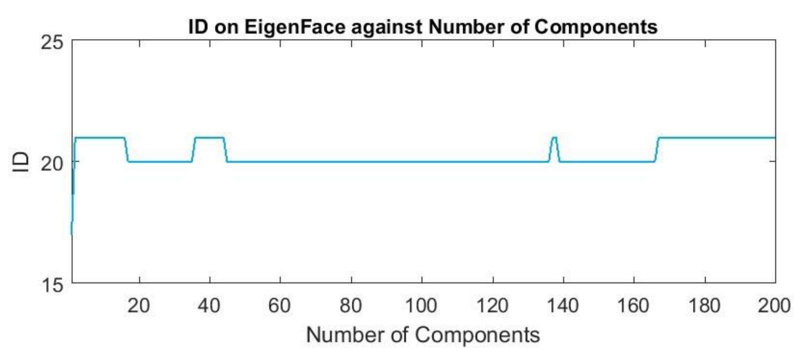


Figure 5.1(b): The ID from the face recognizer changes between two classes of the same person

Confidence is plotted in figure 14, increasing with components. From this plot it appears the best is when components are below 20.

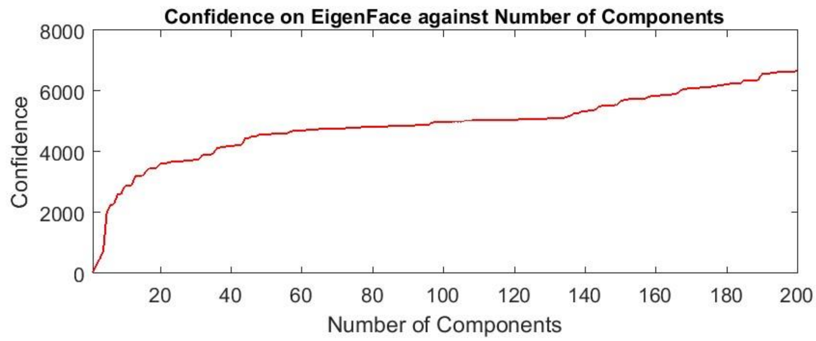


Figure 5.1(c): The Confidence increasing with No. of components[21]

The ID results from Fisherface are more stable than Eigenface and is on ID-21 as seen in .

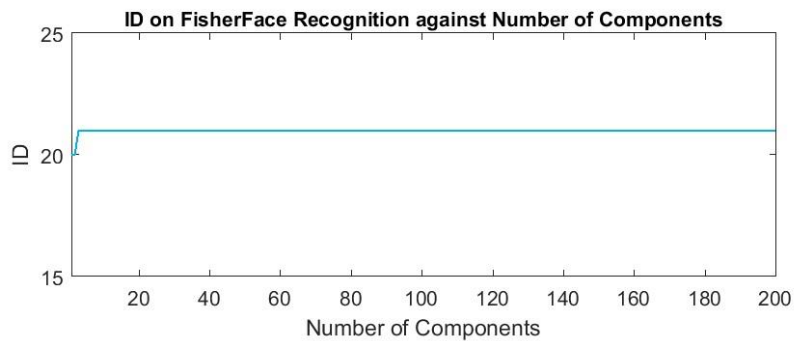


Figure 5.1(d): Graph for stable Fisherface ID[22]

Figure 5.1(d) shows a growth in Fisherface confidence until the number of components reaches ten, which will be utilised as the optimum value. There are several parameters that may be altered in LBPH. All are raised to the maximum.

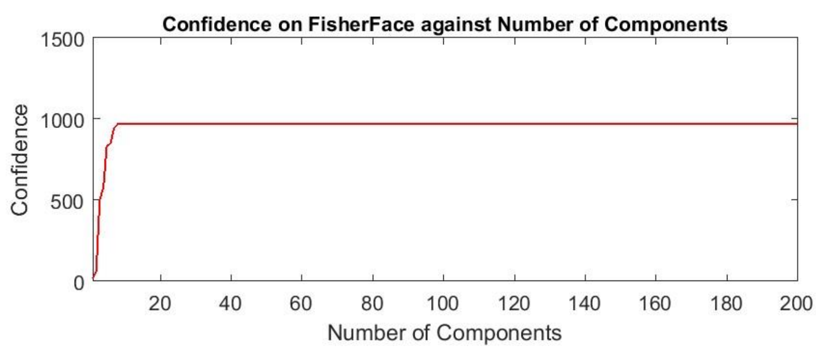


Figure 5.1(e): Stable confidence after 10 components[23]

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 6.5.

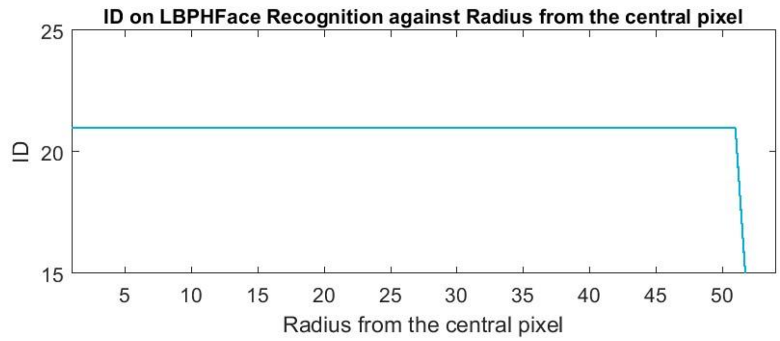


Figure 5.1(f): The ID returned from LBPH[24]

Confidence level is graphed against the radius in figure 6.6. The confidence is fluctuating after 40. The lowest confidence level is at 2.

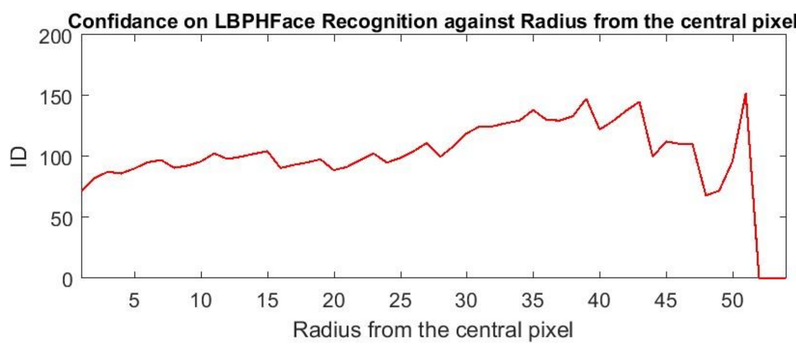


Figure 5.1(g): The confidence returned from LBPH[25]

The number of neighbors is changed from 1 - 13. Further increase caused the computer to stall. The returned ID is plotted below in figure 6.7. ID steady until 9 neighbors and changed to ID-20.

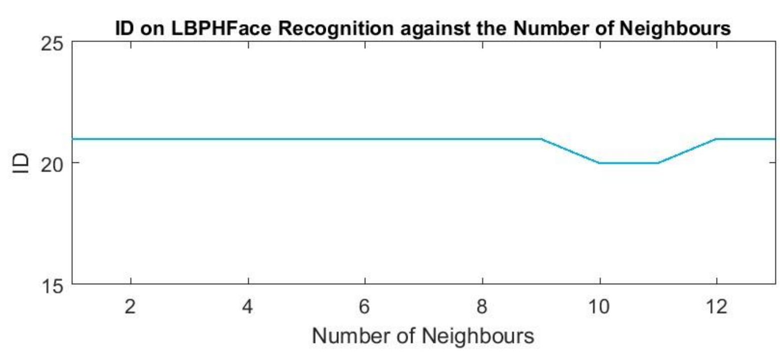


Figure 5.1(h): The ID returned from LBPH changing neighbors[26]

The confidence continuously increased as can be seen in figure 6.8 and 1 neighbors will be included to the next test.

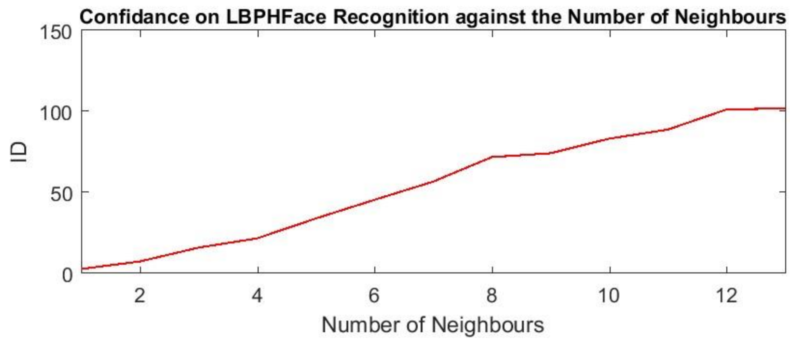


Figure 5.1(i): The Confidence returned from LBPH changing neighbors[27]

The number of cells in X and Y directions are changed simultaneously . ID return is plotted below in figure 6.9. The ID changed from ID-20 to ID -21 and stay steady.

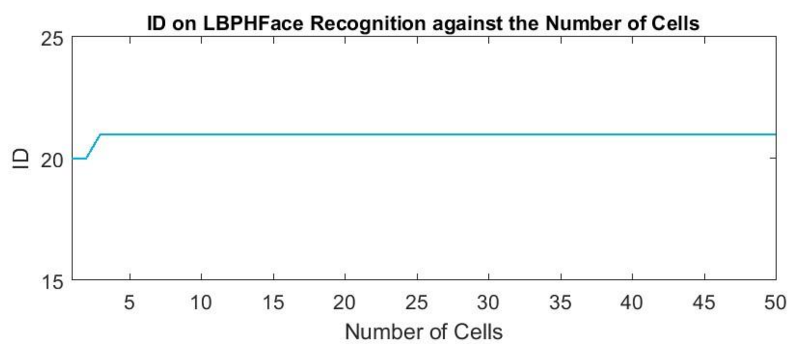


Figure 5.1(j): The ID returned from LBPH changing the number of cells[28]

The returned confidence is plotted in figure 6.10. The confidence was low when cells were less than 8 per side and increased rapidly after 10 ending up more than 1700 at 50.

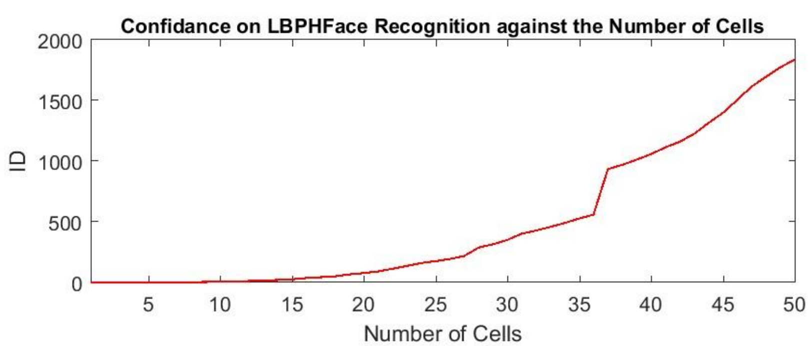


Figure 5 .1(k): The confidence returned from LBPH changing the number of cells[29]

Chapter 6
Conclusion and Future work

6.1 Limitations of facial recognition technology

The Boston Marathon explosions exposed to the broader public the limits of facial-recognition technology. Many private citizens, accustomed to seeing computers on television and in movies match photographs to motor vehicle and other databases in seconds, were surprised that our nation's top law enforcement agencies did not have the same level of technological sophistication available to them when Boston's, and possibly the country's, security was threatened.

Since 9/11, the federal government has invested millions of dollars on facial-recognition technology, with subsidies totaling millions of dollars flowing to state and municipal governments for database development. Despite the fact that government databases held photos of both Boston suspects, technology was unable to match surveillance footage to database images.

Before addressing the limitations of today's technology, let's discuss how one type of facial recognition technology works.

The detection of faces comes first. Typically, the algorithms cycle through several boxes seeking for faces of a specific dimension. Within those boxes, the algorithm recognises facial landmarks and provides a score, indicating a level of confidence that the image is a face. When a face is verified to be a face, the technology produces a template based on parameters such as the distance between the eyes, the area immediately under the nose and above the mouth, and ear to ear.

The generated mathematical representation is then compared to other identified faces. The similarity in ratios between distances on different points of the face, generally concentrated around anchors such as the nose, eyes, ears, and mouth, produces a logarithmic scale score. Close matches have a score of 3 to 5, and definitive nonmatches have a score of less than 1. A score of 40+ is possible when the same picture serves as both probe and target.

Several factors limit the effectiveness of facial-recognition technology:

1. Image Quality

The quality of an image influences how effectively facial-recognition algorithms perform. When compared to a digital camera, the visual quality of scanning video is relatively low. Even high-definition video is just 1080p (progressive scan) at best; most of the time, it is 720p. These numbers correspond to about 2MP and 0.9MP, respectively, but a low-cost digital camera achieves 15MP. The distinction is pretty obvious.

2. Image Size

When a face-detection algorithm detects a face in an image or a still from a video clip, the relative size of that face in relation to the total picture size influences how effectively the face is recognised. Because of the small picture size and the target's distance from the camera, the identified face is only 100 to 200 pixels on a side. Furthermore, scanning an image for www.studymafia.org changing face sizes is a processor-intensive task. Most algorithms enable you to choose a facesize range to assist minimise false positives and speed up picture processing.

3. Face Angle

The relative angle of the target's face influences the recognition score profoundly. When a face is enrolled in the recognition software, usually multiple angles are used (profile, frontal and 45- degree are common). Anything less than a frontal view affects the algorithm's capability to generate a template for the face. The more direct the image (both enrolled and probe image) and the higher its resolution, the higher the score of any resulting matches.

4. Processing And storage

Even while high-definition video has a much lower resolution than digital camera pictures, it still takes up a lot of disc space. Because processing every frame of video is a massive task, just a subset (10% to 25%) is generally processed through a recognition system.
Agencies can utilise computer clusters to reduce total processing time. However, adding computers necessitates significant data transfer via a network, which may be constrained by input-output constraints, further reducing processing speed.

Surprisingly, people outperform technology when it comes to facial recognition. However, while watching a source video, people can only look for a few persons at a time. A computer can compare a large number of people against a database of thousands.

As technology improves, higher-definition cameras will become available. Computer networks will be able to move more data, and processors will work faster. Facial-recognition algorithms will be better able to pick out faces from an image and recognize them in a database of enrolled individuals. The simple mechanisms that defeat today's algorithms, such as obscuring parts of the face with sunglasses and masks or changing one's hairstyle, will be easily overcome.

An immediate way to overcome many of these limitations is to change how images are captured. Using checkpoints, for example, requires subjects to line up and funnel through a single point. Cameras can then focus on each person closely, yielding far more useful frontal, higher resolution probe images. However, wide-scale implementation increases the number of cameras required.

Biometric applications that are evolving are promising. They include movements, expressions, gait and vascular patterns, as well as iris, retina, palm print, ear print, speech recognition, and smell signatures. A mix of modalities is preferable since it increases a system's ability to deliver findings with more certainty. Associated efforts are aimed towards increasing the ability to acquire information from a distance while the target is passive and frequently unaware.

Clearly, there are privacy issues around this technology and its application. Finding a balance between national security and people' private rights will be a hot topic in the coming years, especially as technology advances.

6.2 Advantages And Disadvantages

Facial recognition technology is a relatively new method of identifying persons who may be hazardous or need to be tracked down. It works by selecting faces out of a crowd, taking the required dimensions, and comparing them to photos already in its database.

Advantages:

- Can prevent card counters, etc. from entering casinos
- Can identify terrorists, criminals, etc.
- Can find missing children
- Prevents voter fraud
- Targets shoppers

Disadvantages:

- Isn't always accurate
- Hindered by glasses, masks, long hair etc.
- Must ask users to have a neutral face when pictures are being taken
- Considered an invasion of privacy to be watched

6.3 Summary :

In this project, we built a face detection and recognition system using python OpenCV. We used the face recognition library to perform all the tasks. We've learned about how the face detection system works and how the face recognition system works through this project The most basic task on Face Recognition is of course, "Face Detecting". Before anything, you must "capture" a face (Phase 1) in order to recognize it, when compared with a new face captured on future (Phase 3).
The most common way to detect a face (or any objects), is using the "Haar Cascade classifier"

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. The good news is that OpenCV comes with a trainer as well as a detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one.

6.4 Future work

The future of facial recognition technology is bright. Forecasters opine that this technology is expected to grow at a formidable rate and will generate huge revenues in the coming years. Security and surveillances are the major segments which will be deeply influenced. Other areas that are now welcoming it with open arms are private industries, public buildings, and schools. It is estimated that it will also be adopted by retailers and banking systems in coming years to keep fraud in debit/credit card purchases and payment especially the ones that are online. This technology would fill in the loopholes of largely prevalent inadequate password system. In the long run, robots using facial recognition technology may also come to foray. They can be helpful in completing the tasks that are impractical or difficult for human beings to complete.

References

1] Takeo Kanade. Computer recognition of human faces, volume 47. Birkh¨auser Basel, 1977.

[2] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. Josa a, 4(3):519–524, 1987.

[3] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, Jan 1991.

[4] Dong chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. IEEE Transactions on Geoscience and Remote Sensing, 28(4):509–512, Jul 1990.

[5] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In 2009 IEEE 12th International Conference on Computer Vision, pages 32–39, Sept 2009.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711–720, Jul 1997.

[7] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 1, pages I–511–I–518 vol.1, 2001.

[8] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. JOSA A, 2(7):1160–1169, 1985.

[9] S Marˆcelja. Mathematical description of the responses of simple cortical cells. JOSA, 70(11):1297– 1300, 1980.

[10] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–I. IEEE, 2002.

[11] John P Lewis. Fast template matching. In Vision interface, volume 95, pages 15–19, 1995.

[12] Meng Xiao He and Helen Yang. Microarray dimension reduction, 2009.
[13] L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in IEEE Access, vol. 8, pp. 139110-139120, 2020, doi: 10.1109/ACCESS.2020.3011028.

[14] https://ieeexplore.ieee.org/abstract/document/9145558

[15] https://arxiv.org/ftp/arxiv/papers/1302/1302.6379.pdf

[16] ttps://www.researchgate.net/publication/233864740_Face_Recognition_A_Literature_Review

[17] https://iopscience.iop.org/article/10.1088/1755-1315/170/3/032110/pdf

[18]https://thesai.org/Downloads/Volume9No6/Paper_6-study_of_Face_Recognition_Techniques.pdf

[19] https://core.ac.uk/download/pdf/26858086.pdf

[20]https://www.ijert.org/research/a-survey-on-various-problems-challenges-in-face-recognition-IJERTV2IS60850.pdf