A Thesis/Project/Dissertation Report

on

FACE DETECTION AND RECOGNIZATION SYSTEM

Submitted in partial fulfillment of the

requirement for the award of the degree of

Computer Science and Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr.P Muthusamy
Professor

Submitted By

Radheyshyam Kumar Verma
19SCSE1010645
Piyush Anand
19SCSE1010679

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DEC, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Face detection and recognization system "** in partial fulfillment of the requirements for the award of the Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the Sept 2021, to Dec 2021, under the supervision P Muthusamy, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Radheyshyam Kumar Verma,19SCSE1010645
Piyush Anand,19SCSE1010679

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr P.Muthusamy
Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Name: Radheyshyam Kumar Verma 19SCSE1010645, Piyush Anand 19SCSE1010679 has been held on ________________ and his/her work is recommended for the award of Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December,2021
Place: Greater Noida

Acknowledgement

Abstract

Facial Recognition represents the event of a system which may determine the person with the assistance of a face using Computer Vision (Open CV). Face recognition is employed within the fields of Identity Recognition, police investigation and enforcement. It's a method of characteristic someone supported facial expression. This method is enforced in 2 stages. They're the training stage and therefore the testing stage. This study primarily consists of 3 elements, specifically face detection from the image, feature extraction and storing many reminder images, and recognition. Face finding rule is employed to detect the face from the given image. The foremost helpful and distinctive options of the face image are extracted within the feature extraction part. Face Detection may be challenging because of pictures and video frames will contain advanced background, completely different head poses and occlusion like carrying glasses or scarf. It presents a rule for finding face recognition downside and concatenated into one feature vector that is employed to coach the system to recognise among the prevailing photos with it. Within the testing stage the system takes the face of the image of someone for recognition. Image acquisition, preprocessing, image filtering, feature extraction is just like the learning stage. For classification the options are fed to the trained system. The algorithms can determine the face image from the content and acknowledges it.

The growing interest in computer vision of the past decade. Fueled by the steady doubling rate of computing power every 13 months, face detection and recognition has transcended from an esoteric to a popular area of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. This application will be developed using Intel’s open source computer vision project, OpenCV. Over the past decade face detection and recognition have transcended from esoteric to popular areas of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. A general statement of the face recognition problem (in computer vision) can be formulated as follows: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces. Facial recognition generally involves two stages: Face Detection where a photo is searched to find a face, then the image is processed to crop and extract the person’s face for easier recognition. Face Recognition where that detected and processed face is compared to a database of known faces, to decide who that person is.

Contents

List of Figures

CHAPTER-1 Introduction

1.1 Introduction

Face recognition from image or video is a popular topic in biometrics research.Many public places usually have surveillance cameras for video capture and these cameras have their significant value for security purpose. It is widely acknowledged that the face recognition have played an important role in surveillance system as it doesn’t need the object’s cooperation. The actual advantages of face based identification over other biometrics are uniqueness and acceptance. As human face is a dynamic object having high degree of variability in its appearance, that makes face detection a difficult problem in computer vision. In this field, accuracy and speed of identification is a main issue.

The goal of this project is to evaluate various face detection and recognition methods, provide complete solution for image based face detection and recognition with higher accuracy, better response rate as an initial step for video surveillance. Solution is proposed based on performed tests on various face rich databases in terms of subjects, pose, emotions, race and light.

Nowadays advancement of man-made brainpower is effectively creating; they open up tremendous potential outcomes before us. Investigation, gauging, detection went to another level with the utilization of man-made reasoning advancements. As of late, an incredibly encouraging field of research is Computer vision. Face detection is a phase where identifying the faces from the images or video sources. It very well may be utilized for remote distinguishing proof administrations for security in regions, for example, banking, transportation, law requirement, and electrical businesses. Despite huge varieties in visual upgrades because of evolving condition, maturing, and interruptions like whiskers, glasses, and haircut changes, this capacity is extremely powerful. In this project we proposed a numerical model and computational model of face acknowledgment which is quick, sensibly basic, and precise in compelled condition.. Face detection utilizing Eigen face has been demonstrated to be precise and quick. With the guide of a normal web camera, a machine can identify and perceive an individual's face; a custom login screen with the capacity to channel client get to dependent on the clients' facial highlights will be created. The goals of this is to give a lot of location calculations that can be later bundled in an effectively compact structure among the distinctive processor designs we find in machines today. This will examine PC vision, to be specific face location based Open CV library.

Computer Vision is a very new area of research is currently hanging in the Computer Science under. It mainly want to solve the problem is to build an intelligent system, extracted from the images useful information, of course, we could simply say, the ultimate purpose of computer vision is to create such as like humans can be made for the image the wisdom of response and identification systems. The following chart is the relation between different fields.

Face Recognition is one of the more representative and classic application in Computer Vision. Face Detection is the most important step of face recognition. Not only the face recognition, face detection also a first step in Human Computer Interaction (HCI) systems. E.g. expression recognition. Unlike traditional HCI device, keyboard, mouse and display, it provides more effective methods to increase user experience with computer used. As a result, it speeds up human’s work. It’s conveyed information from physics world into logical thinking to control the computer system. In additional, face detection is one of an object detection which to classify the desired object from the given images/video and located it. License Plate Detection is one of examples about the object detection. In biometric approaches, human faces are unique object like fingerprints, iris which widely used in security issues. Many types of personal authentications system have been developed related this approaches which takes advantage of unique and special characteristics. System can be searched effectively to screen out useful information (face) from dozes of video media or photos from internets.

For example, Video surveillance in UK, there is one CCTV cameras for every 14 people.[2] They also need to analyze all these video, which use face detection to extract any useful information and store it for further used.

Human faces are non-rigid objects and appeared in different scale, pose, angle and facial expressions. Human faces always have variations for example, glasses. In additional, the images have different brightness, contrast. These results are the challenge of the face detection.

The aim of this project is to develop and propose a system to detect human faces in digital images effectively, no matters what person’s ethnic, pose. Input images may be varied with face size, complex of background and illumination condition.

Also, cross-platform is preferred because of it can be worked not only the windows-platform to increase the utilization.

1.2 Problem Formulation

Face recognition is an easy task for humans. Experiments in have shown, that even one to three day old babies are able to distinguish between known faces. So how hard could it be for a computer? It turns out we know little about human recognition to date. Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? How do we analyze an image and how does the brain encode it? It was shown by David Hubel and Torsten Wiesel, that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don’t see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them.

Face recognition based on the geometric features of a face is probably the most intuitive approach to face recognition. One of the first automated face recognition systems was described in marker points (position of eyes, ears, nose, ...) were used to build a feature vector (distance between the points, angle between them, ...). The recognition was performed by calculating the euclidean distance between feature vectors of a probe and reference image. Such a method is robust against changes in illumination by its nature, but has a huge drawback: the accurate registration of the marker points is complicated, even with state of the art algorithms. Some of the latest work on geometric face recognition was carried out in. A 22-dimensional feature vector was used and experiments on large datasets have shown, that geometrical features alone my not carry enough information for face recognition.

The Eigenfaces method described in took a holistic approach to face recognition: A facial image is a point from a high-dimensional image space and a lower-dimensional representation is found, where classification becomes easy. The lower-dimensional subspace is found with Principal Component Analysis, which identifies the axes with maximum variance. While this kind of transformation is optimal from a reconstruction standpoint, it doesn't take any class labels into account. Imagine a situation where the variance is generated from external sources, let it be light. The axes with maximum variance do not necessarily contain any discriminative information at all, hence a classification becomes impossible. So a class-specific projection with a Linear Discriminant Analysis was applied to face recognition in. The basic idea is to minimize the variance within a class, while maximizing the variance between the classes at the same time.

Recently various methods for a local feature extraction emerged. To avoid the high-dimensionality of the input data only local regions of an image are described, the extracted features are (hopefully) more robust against partial occlusion, illumation and small sample size. Algorithms used for a local feature extraction are Gabor Wavelets, Discrete Cosinus Transform and Local Binary Patterns. It's still an open research question what's the best way to preserve spatial information when applying a local feature extraction, because spatial information is potentially useful information. Over the past decade face detection and recognition have transcended from esoteric to popular areas of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. Because of the intrinsic nature of the problem, computer vision is not only a computer science area of research, but also the object of neuroscientific and psychological studies also, mainly because of the general opinion that advances in computer image processing and understanding research will provide insights into how our brain work and vice versa. A general statement of the face recognition problem (in computer vision) can be formulated as follows: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces. Facial recognition generally involves two stages: Face Detection where a photo is searched to find a face, then the image is processed to crop and extract the person’s face for easier recognition. Face Recognition where that detected and processed face is compared to a database of known faces, to decide who that person is. Since 2002, face detection can be performed fairly easily and reliably with Intel’s open source framework called OpenCV.This framework has an inbuilt Face Detector that works in roughly 90-95% of clear photos of a person looking forward at the camera. However, detecting a person’s face when that person is viewed from an angle is usually harder, sometimes requiring 3D Head Pose Estimation. Also, lack of proper brightness of an image can greatly increase the difficulty of detecting a face, or increased contrast in shadows on the face, or maybe the picture is blurry, or the person is wearing glasses, etc. Face recognition however is much less reliable than face detection, with an accuracy of 30-70% in general. Intel's open-source computer-vision library can greatly simplify computervision programming. It includes advanced capabilities - face detection, face tracking, face recognition, Kalman filtering, and a variety of artificialintelligence (AI) methods - in ready-touse form. In addition, it provides many basic computer-vision algorithms via its lower-level APIs.

1.3 Tools and Technology used

The 3 main modules that are required for our research are: 1. OpenCV 2. NumPy 3. Simple CV

There are several Python libraries related to Image Processing and Computer Vision. The one that will be presented in this report are: 1. Numpy 2. SimpleCV 3. OpenCV 4. Python

3.1 Numpy: It is a general-purpose package which is used for array processing. It is a fundamental package that combines with python and can be used for scientific computing. Numpy reduces the man’s works by helping them in integrating a wide variety of databases with the help of arbitrary data-types. It contains various features including a powerful n- dimensional array object, sophisticated(broadcasting)functions, tools for integrating C/C++ and Fortran code and useful linear algebra and random number capabilities.

3.2 SimpleCV: It is an open-source framework i.e. it is a collection of packaged and libraries that are used for image manipulation. First of all how a computer reads an image? Basically, for the color, you have 3 channels(popularly known as RGB) and if at all you want your image black and white then you have only one channel. In RGB format each element represents the intensity of the brightness of the pixel so basically computer will read an image in the form of the matrix. If you want colored image then it will import read in 3d matrix and for black and white format it will read in 2d matrix. First, you have to import a cv2 module in your python library. If you are using idle then with the help of command prompt you can install cv2 and if Pycharm(Python's Integrated Development Environment developed by JetBrains) then you can install it through settings. Imread() will read image with the help of its parameters. First of all you have to give the image file name or source path of the image that you want to display. You can also change the image appearance into grayscale or colored format. For grayscale image you will write 0 and for colored image, you will write 1. Based upon that the matrix will be printed in 2d and 3d format.

3.3 OpenCV: OpenCV is by far the most capable and most commonly used computer vision library. It supports a wide variety of languages like Python, C/C++, Java, etc. In SimpleCV we have printed the image in a matrix format. Now with the help of OpenCV, we will print the actual image and it is done by the imshow() method. If you want an image in your original size you can skip that step and directly write img instead of resize). Imshow() method is used for displaying your image. waitKey() method is used for displaying the image for specified milliseconds. If you write 0, as soon as you press any key it will destroy that window. If you write 25, it will display the frame image for 25milliseconds after which the window gets destroyed with the help of destroyAllWindows() method.
OpenCV is a synonym of Open Computer Vision Library, which has at least 500 algorithms, documentation and sample code for real time computer vision.[7] OpenCV is originally developed by Intel and launched in 1999. It free for commercial and research used. [8] OpenCV library is cross-platform which means its can execute on Windows, Mac OS X, Linux, PSP and other embedded devices.

The library is mainly written in C, which makes it easily possible to transfer into specific platforms.

Example application of OpenCV library is Human-Computer Interaction, Object Identification, Segmentation and Recognition and so on.[9] Stanley was implemented by OpenCV, which was the winning entry to the 2005 DARPA Grand Challenge race.

There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of face dimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing.

3.4 Python:-Python is high-level programming language like a Perl, Ruby and Tcl which are used as a scripting language. It was conceived by Guido van Rossum in 1989. Also, python is one of the three “official languages” in Google which means that more application in Google was deployed this language. E.g. Google App Engine SDK. Here are the points that make python is selected:

1. Free, Python is product of open source. People allows to use it in business or commercial without any charge.
2. Easy to read, Syntax in Python is clear and readable. Beginner can be easily to read and handle Python’s coding very well.
3. Rapid development, it is because it likes pseudo code. Everything coding in Python is direct result.
4. Highly portability, Python is working on different platforms, because of Python is written portable ANSI C.
5. Reusability, Python is easily reused modules and packages. Peoples can be developed their own library and reused it later project.
6. Object-Oriented Programming. Unlike scripting language, Python is designed to be object-oriented. OO programming means you can implemented using idea of inheritance and polymorphism.

Hardware and Software requirements

Hardware

This system has been run and tested on following configuration. In order to get better performance, update RAM to 2GB or more is recommended.

Here is a list of minimum hardware requirements:

1. Intel Pentium Processor or any other compatible Processor, 1 GHz or greater
2. Minimum 128 MB of RAM capacity or more
3. Minimum of 32 MB Graphic Card RAM capacities or more
4. Recommended hard disk space of 1GB or more.

Software

This system was developed by cross-platform component. E.g. python and pyqt4. It means that face detection system can be operated in windows-platform, mac system, Linux-related system. These software tools will be discussed in later section.

Here is a list of software requirements:

1. Windows 2000/XP or above, Linux. Mac OS X, Unix
2. Python 2.5

3. PyQt-Py2.5

1.4 Working Methodology

Face recognition takes a photo from a video or a digital camera as input and outputs the diagnosed photo subject matter. Facial features may additionally consist of regions inside the face, variations within the face structure, face cuts and angles which have been formatted and styled. Face extraction includes grabbing of the capabilities from camera. Face detection includes the elimination of the background and focusing on the foreground eliminating some other elements apart from the face vicinity, but the device nevertheless pertains some drawbacks because it cannot come across the head be counted which can be a gift because of overlapping of faces or mistaken recognition of faces having similar facial functions.

•Find faces - regardless of whether the errand of perceiving individuals in photos, or video acknowledgment, or whatever else.

•Face positioning - pics aren't regularly located on which an individual stand straightforwardly before the focus, often the face grows to become, we are facing the challenge of situating it as though the picture become taken legitimately.

•Defining outstanding facial capabilities - this development can be referred to as a full face acknowledgment step, it examinations the photograph and gets certainly one of a type automated estimations of the face.

•Identification of a person - we assessment a got information and the information efficiently accessible to us, if the statistics are similar, we will show the call of the character, if now not, in like way we've now not recognised at this point to us character. This will analyse in element each one of the means to manufacture a face acknowledgment framework and comparison their execution and the help of various libraries, simply as the velocity of crafted by way of every section in various libraries of Computer vision.

In general, FD can be implemented by four methods: knowledge based methods, template matching, invariant feature methods and learning based methods. These methods will be introduced with the following:

1. Knowledge based methods: The models are used human knowledge to find a face patterns from the testing images. Based on the nature of human faces, algorithms scan the image from top-to-bottom and left-to-right order to find facial feature. For instance, face should be including two eyes and mouth…

Pros: Easy applied into simple rules

Cons: difficult to detect in invariant background, such as different pose, uncontrolled illumination. Well results based on well-defined rules. This algorithm does not work on the pose.

2. Template marching: The model is used several templates to find out the face class and extract facial features. Rules are pre-defined and decide whether there is face in the image. For instance, using filters to extract the contours of face shape

Sample of template marching
Pros: Simple to apply this method.
Cons: similar to knowledge based method, hard to detect face in different poses. Algorithms are sensitive to scale size, face shape and pose.

3. Invariant feature methods: The model is bottom-up approaches and used to find a facial feature (eyebrows, nose), even in the presence of composition, perspective vary, so it is difficult to find a face real time using this method. Statistical models are developed to determine the faces. Facial features of human faces are: shape, texture, skin.

Pros: Unlike knowledge-based method, it is invariant to pose and expression. Cons: not suitable to detect facial features from uncontrolled background, time consuming algorithms. Detection rate is not accuracy, because of need to combine different feature and processing it.

4. Learning based methods: The models are trained from a set of training set before doing detection. For the large amount of training data, it can be provided high accuracy recognition rate to resist variation, expression and pose of faces images. For instance, Many of “non-face” and “face” images import into the system. Machine learning techniques are employed to train the system based on

the statistical properties and probability distribution function. Principle Component Analysis (PCA), Support Vector Machine (SVM), Naïve Bayes Classifier, Hidden Markov model, Neural Network and Adaboost are well-known classifiers to use for face detection.

Pros: fast to detect face. Can be detected different pose and orientation if have enough training set. Showed a good empirical results.

Cons: need more and more “non-face” and “face” sample for training, need to scan different scale.

CHAPTER-2 Literature Review/Project Design

2.1 Literature Survey

In this section, face detection methods will be introduced with short overview to give any general ideal about the history of face detection and the future approaches of it.In the present, advances of PC vision are effectively creating, with their assistance, we can tackle issues all the more adequately, one of which is acknowledgment. Because of dynamic improvement, designers get countless libraries to take care of issues related with PC vision. Works center around the hypothetical parts of building a steady framework for face acknowledgment. This report portrays the genuine strategies and advances for all phases of the improvement of the acknowledgment framework, since in the field of acknowledgment, an enormous number of one of a kind arrangements have been created. This paper is a utilization of face identification and following in recordings and cameras utilizing Open CV. Face recognition has been a strong field of research since the 1990s, but is still a far way away from a reliable method of user authentication. More and more techniques are being developed each year. The Eigenface technique is considered the simplest method of accurate face recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate. OpenCV was started at Intel in 1999 by Gary Bradski for the purposes of accelerating research in and commercial applications of computer vision in the world and, for Intel, creating a demand for ever more powerful computers by such applications. Vadim Pisarevsky joined Gary to manage Intel's Russian software OpenCV team. Over time the OpenCV team moved on to other companies and other Research. Several of the original team eventually ended up working in robotics and found their way to Willow Garage. In 2008, Willow Garage saw the need to rapidly advance robotic perception capabilities in an open way that leverages the entire research and commercial community and began actively supporting OpenCV, with Gary and Vadim once again leading the effort. It is a PC development which is being used in a grouping of employments that recognizes human faces in cutting edge pictures. Face identification is one of the most discussed in innovation. Face restriction can be alluded to as extraction of facial highlights utilizing design acknowledgment framework. Both MATLAB and Open CV can be utilized for execution. Face Detection and Recognition using Open cv Based on Fisher faces Algorithm. Face Detection is a technology to determine human face in videos and arbitrary (digital) images. The aim of face detection is detect faces in any images or videos. If yes, it's reported the location on the images or videos. It just only concerns facial features in testing images. In other words, face detection can be regarded as a pattern classification. Face detection is the first part of the face recognition, it is because we need to clarify facial feature before doing recognition step.

Face detection can be regarded as a specific case of object-class detection. For the detection is to locate the face in the digital images/video stream, no matter what the pose, scale, facial expressions. In other words, face detection algorithms to handle pattern classification. It task to identify a given images to decides it has face or not.

The following items are the difference between face detection and other face processing :

1. face detection: To determine any face in given images/ video
2. face localization: To locate the location of the face in given images/ video
3. face recognition: To recognize faces in given images/ video

History of face detection

In the early stage, face detection algorithms mainly focused to detect the frontal human face. However, newer algorithms try to consider the different view of face as a core of face detection.

The first of face detection system has been developed since in early 1970’s.[14] Due to the limitation of computation, system can’t be satisfied the requirement of users, which is identify passport photograph real time.

At the beginning of 1990’s, techniques are proposed focused on the face recognition on and increase the need of face detection. Many system were constructed to deal with video streaming. In the past few years, lots of methods are developed at least more than 150 methods.

Important for Face Detection

Face detection is not just classified the target object (face) in the images, because of lot of variation, pose variation, facial occlusion. The challenges of face detection discuss on later section of this chapter.

However, the following items are explained how the important of face detection:

1. Face recognition: Face recognition consists in the meaning of identification and verification of the people. These considerations are the security issues.
2. Human Computer Interaction (HCI): Human Computer Interaction is the study of interaction between human and computer.[15] It is important that it is increased the user experience with machine, as a result intelligent human computer interaction system is construct. For example, facial expression recognition, helping the disabled people.
3. Facial Expression Recognition: This technique try to figure out the meaning of expression from detected people.

The following are the example application use of the face detection:

Typical Application	Research Area
Immigration Management	Public Security
Monitoring sensitive characters (spies, terrorists, etc.)	
Automated Login system	Human Computer Interaction (HCI)
Realistic virtual games	
E-commerce authentication	Financial Security

The following research areas are the summary of Face detection/recognition:

1. Biometrics
2. Human Computer Interaction
3. Facial Image coding/ compression
4. Facial Expression analysis
5. Emotional computing
6. Face attribute classification
7. Attractiveness judge

Although many of the algorithms have been proposed and improved within a few years. Face detection are still limited by some uncollected factors. E.g. variant pose, illumination condition, expression, hairstyle and different camera equipment. Either one of the problems can decrease the performance of face detection. At the worst case, it should be the combination of the above problem. However, this situation often happens in our real worlds. E.g. surveillance with low quality.

In addition, occlusion is another of factor to effect the result of the FD rate. E.g make-up, sunglasses. In other words, occlusion sometime will be greatly changed the appearance of faces.

The following items are the summary of the main challenge in face detection:

1. Variant pose: Variant pose is occurred because of peoples not always orient to camera.
2. Illumination condition: Different lighting and the quality of camera directly affect the quality of face. Sometimes it can be varied greater than facial expression and occlusion.
3. Facial expression: Different expression in the face is presented different information to the machine. Face is non-rigid objects which is changed by different expression.
4. Occlusion: Face detection not only deals with different faces, however, it need deal with any optional object. E.g. Hairstyle, sunglasses are the example of occlusion in face detection. For global feature, occlusion is one of major difficulty factor in face detection.
5. Uncontrolled background: Face Detection systems are not only detected faces on uniform environment. In reality, Peoples are always located on complex background with different texture and object. These ‘thing’ are the major factors to effect the performance of face detection system.

The following photos are the example of uncollected factors.

1. Different pose of the same person

2. Same person with different illumination

3. Different facial expressions.

4. Example of occlusions by object.

2.2 Algorithm Used

(a)Principle Component Analysis (PCA)

PCA is a method in which is used to simplify the problem of choosing the representation of eigenvalues and corresponding eigenvectors to get a consistent representation. This can be achieved by diminishing the dimension space of the representation. In order to obtain fast and robust object recognition, the dimension space needs to be reduced. Moreover, PCA also retains the original information of the data. Eigenface based algorithm applies the PCA basis.

(i)Eigenface based algorithm

Eigenface based approach is the most widely used method for face detection. According to Pavanet al., eigenface is well known due to its simplicity, less sensitive in poses and better performance involving small databases or training sets . This approach utilizes the presence of eyes, nose and mouth on a face and relative distances between these objects. This characteristic feature is known as Eigenfaces in facial domain . This facial feature can be extracted by using a mathematical tool called Principle Component Analysis (PCA). By using PCA, any original image from the training set can be reconstructed by combining the Eigenfaces. Generally, a face is classified as a face by calculating the relative distance of the Eigenfaces Since we have a pre-processed facial picture, we can perform Eigenfaces (PCA) for Face Recognition. OpenCV accompanies the capacity "cvEigenDecomposite()", which plays out the PCA activity, anyway we need a database (preparing set) of pictures for it to realize how to perceive every one of your kin. Use "Head Component Analysis" to change over the preparation pictures into a lot of "Eigenfaces" that speak to the fundamental contrasts between the preparation pictures. First it will locate the "normal face picture" of your pictures by getting the mean estimation of every pixel. At that point the eigenfaces are determined in contrast with this normal face, where the first eigenface is the most prevailing face contrasts, and the second eigenface is the second most predominant face contrasts, etc., until you have around 50 eigenfaces that speak to the greater part of the distinctions in all the preparation set pictures.

The problem with the image representation we are given is its high dimensionality. Two-dimensional p \times q grayscale images span a m = pq-dimensional vector space, so an image with 100 \times 100 pixels lies in a 10,000-dimensional image space already. The question is: Are all dimensions equally useful for us? We can only make a decision if there’s any variance in data, so what we are looking for are the components that account for most of the information. The Principal Component Analysis (PCA) was independently proposed by Karl Pearson (1901) and Harold Hotelling (1933) to turn a set of possibly correlated variables into a smaller set of uncorrelated variables. The idea is, that a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The PCA method finds the directions with the greatest variance in the data, called principal components.

→Algorithmic discription

Let X = \{ x_{1}, x_{2},\$ldots$, x_{n} \} be a random vector with observations x_i \in R^{d}.

Compute the mean \mu
\mu = \frac{1}{n} \sum_{i=1}^{n} x_{i}

Compute the the Covariance Matrix S
S = \frac{1}{n} \sum_{i=1}^{n} (x_{i} - \mu) (x_{i} - \mu)^{T}`

Compute the eigenvalues \lambda_{i} and eigenvectors v_{i} of S
S v_{i} = \lambda_{i} v_{i}, i=1,2,\ldots,n

Order the eigenvectors descending by their eigenvalue. The k principal components are the eigenvectors corresponding to the k largest eigenvalues.
The k principal components of the observed vector x are then given by:

y = W^{T} (x - \mu)

where $W = (v_{1}, v_{2}, \ldots, v_{k})$.

The reconstruction from the PCA basis is given by:

$$x = W y + \mu$$

where $W = (v_{1}, v_{2}, \ldots, v_{k})$.

The Eigenfaces method then performs face recognition by:

Projecting all training samples into the PCA subspace.
Projecting the query image into the PCA subspace.
Finding the nearest neighbor between the projected training images and the projected query image.
Still there’s one problem left to solve. Imagine we are given 400 images sized 100×100 pixel. The Principal Component Analysis solves the covariance matrix $S = X X^{T}$, where ${size}(X) = 10000 \times 400$ in our example. You would end up with a 10000×10000 matrix, roughly 0.8 GB. Solving this problem isn’t feasible, so we’ll need to apply a trick. From your linear algebra lessons you know that a $M \times N$ matrix with $M > N$ can only have $N - 1$ non-zero eigenvalues. So it’s possible to take the eigenvalue decomposition $S = X^{T} X$ of size $N \times N$ instead:

$$X^{T} X v_{i} = \lambda_{i} v{i}$$

and get the original eigenvectors of $S = X X^{T}$ with a left multiplication of the data matrix:

$$X X^{T} (X v_{i}) = \lambda_{i} (X v_{i})$$

The resulting eigenvectors are orthogonal, to get orthonormal eigenvectors they need to be normalized to unit length. I don’t want to turn this into a publication, so please look for the derivation and proof of the equations.

(ii) Fisherfaces Algorithm:

The Fisherface approach is also one of the most widely used methods for feature extraction in face images. This approach tries to find the projection direction in which, images belonging to different classes are separated maximally . According to Shang-Hung Lin, Fisherface algorithm is a refinement of the eigenface algorithm to cater the illumination variation . Bulhumeur reported that Fisherface algorithm performs better than eigenface in a circumstance where the lighting condition is varied .This approach requires several training images for each face. Therefore, it cannot be applied to the face recognition applications where only one example image per person is available for training.

The Principal Component Analysis (PCA), which is the center of the Eigenfaces technique, finds a direct blend of highlights that expands the all-out difference in information. While this is plainly an incredible method to speak to information, it doesn't think about any classes thus a great deal of discriminative data might be lost when discarding segments. The Fisherfaces strategy learns a class-explicit change grid, so the they don't catch brightening as clearly as the Eigenfaces technique. The Discriminant Analysis rather finds the facial highlights to separate between the people. It's critical to make reference to, that the exhibition of the Fisherfaces intensely relies upon the info information too. For all intents and purposes stated: on the off chance that you become familiar with the Fisherfaces for wellenlightened pictures just and you attempt to perceive faces in terrible lit up scenes, at that point technique is probably going to locate an inappropriate part (in light of the fact that those highlights may not be overwhelming on awful lit up pictures). This is to some degree consistent, since the technique got no opportunity to get familiar with the brightening.

The Principal Component Analysis (PCA), which is the core of the Eigenfaces method, finds a linear combination of features that maximizes the total variance in data. While this is clearly a powerful way to represent data, it doesn’t consider any classes and so a lot of discriminative information may be lost when throwing components away. Imagine a situation where the variance in your data is generated by an external source, let it be the light. The components identified by a PCA do not necessarily contain any discriminative information at all, so the projected samples are smeared together and a classification becomes impossible (see http://www.bytefish.de/wiki/pca_lda_with_gnu_octave for an example).

The Linear Discriminant Analysis performs a class-specific dimensionality reduction and was invented by the great statistician Sir R. A. Fisher. He successfully used it for classifying flowers in his 1936 paper The use of multiple measurements in taxonomic problems [Fisher36]. In order to find the combination of features that separates best between classes the Linear Discriminant Analysis maximizes the ratio of between-classes to within-classes scatter, instead of maximizing the overall scatter. The idea is simple: same classes should cluster tightly together, while different classes are as far away as possible from each other in the lower-dimensional representation. This was also recognized by Belhumeur, Hespanha and Kriegman and so they applied a Discriminant Analysis to face recognition in.
Let X be a random vector with samples drawn from c classes:

$$\begin{align*}
X & = & \{X_1,X_2,\ldots,X_c\} \\
X_i & = & \{x_1, x_2, \ldots, x_n\}
\end{align*}$$

The scatter matrices S_{B} and S_{W} are calculated as:

$$\begin{align*}
S_{B} & = & \sum_{i=1}^{c} N_{i} (\mu_i - \mu)(\mu_i - \mu)^{T} \\
S_{W} & = & \sum_{i=1}^{c} \sum_{x_{j} \in X_{i}} (x_j - \mu_i)(x_j - \mu_i)^{T}
\end{align*}$$

, where μ is the total mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

And μ_i is the mean of class $i \in \{1,\ldots,c\}$:

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

Fisher’s classic algorithm now looks for a projection W, that maximizes the class separability criterion:

$$W_{opt} = \operatorname{arg\,max}_{W} \frac{|W^T S_B W|}{|W^T S_W W|}$$

Following, a solution for this optimization problem is given by solving the General Eigenvalue Problem:

$$\begin{align*}
S_{B} v_{i} & = & \lambda_{i} S_w v_{i} \nonumber \\
S_{W}^{-1} S_{B} v_{i} & = & \lambda_{i} v_{i}
\end{align*}$$

There’s one problem left to solve: The rank of S_{W} is at most (N-c), with N samples and c classes. In pattern recognition problems the number of samples N is almost always samller than the dimension of the input data (the number of pixels), so the scatter matrix S_{W} becomes singular (see [RJ91]). In [BHK97] this was solved by performing a Principal Component Analysis on the data and projecting the samples into the (N-c)-dimensional space. A Linear Discriminant Analysis was then performed on the reduced data, because S_{W} isn’t singular anymore.

The optimization problem can then be rewritten as:

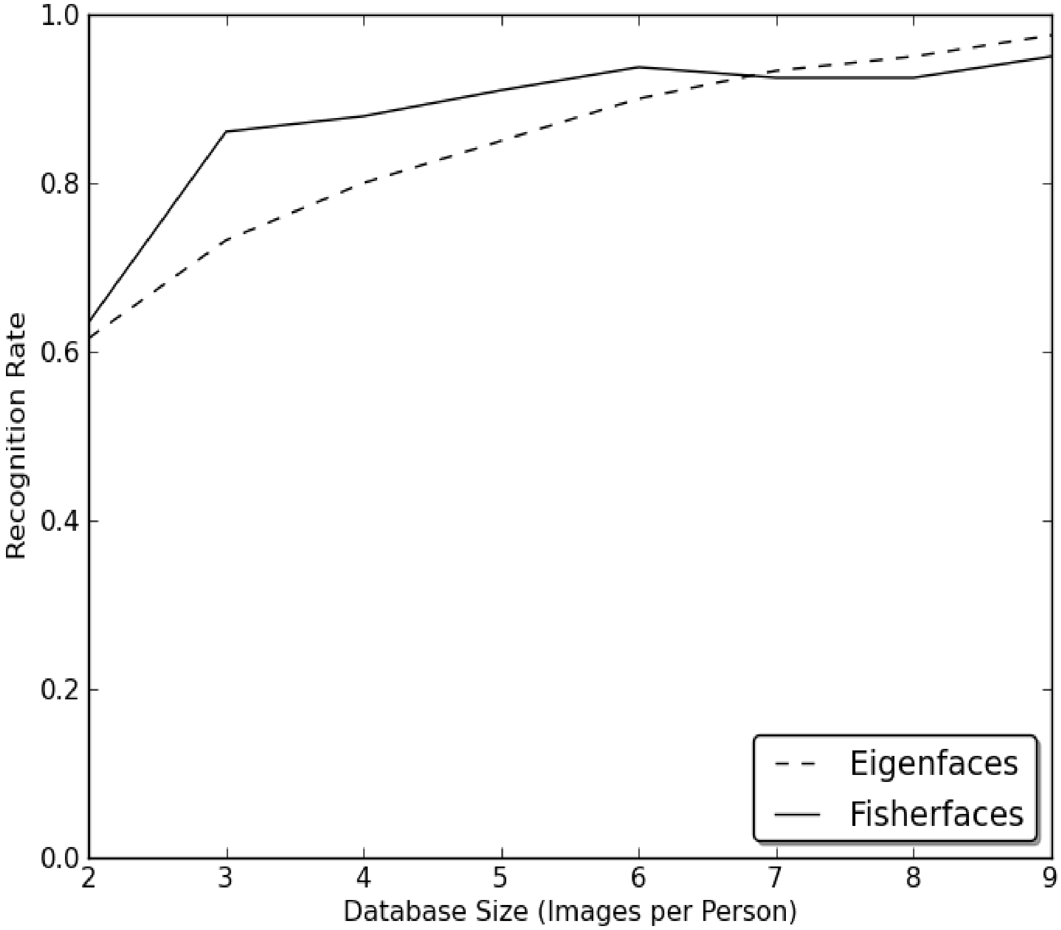
$$\begin{align*}
W_{pca} & = & \operatorname{arg\,max}_{W} |W^T S_T W| \\
W_{fld} & = & \operatorname{arg\,max}_{W} \frac{|W^T W_{pca}^T S_{B} W_{pca} W|}{|W^T W_{pca}^T S_{W} W_{pca} W|}
\end{align*}$$

The transformation matrix W, that projects a sample into the (c-1)-dimensional space is then given by:

$$W = W_{fld}^{T} W_{pca}^{T}$$

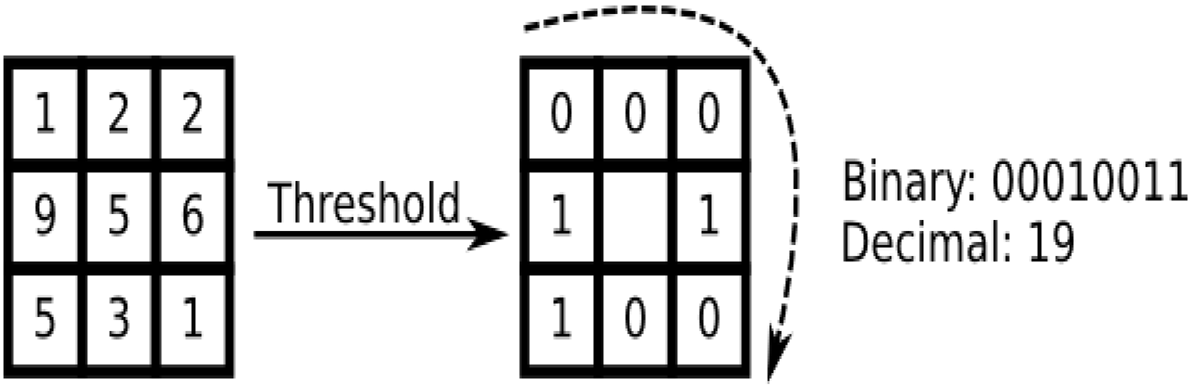
Eigenfaces and Fisherfaces take a somewhat holistic approach to face recognition. You treat your data as a vector somewhere in a high-dimensional image space. We all know high-dimensionality is bad, so a lower-dimensional subspace is identified, where (probably) useful information is preserved. The Eigenfaces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components with a maximum variance over all classes aren't necessarily useful for classification (see http://www.bytefish.de/wiki/pca_lda_with_gnu_octave). So to preserve some discriminative information we applied a Linear Discriminant Analysis and optimized as described in the Fisherfaces method. The Fisherfaces method worked great... at least for the constrained scenario we've assumed in our model.

Now real life isn't perfect. You simply can't guarantee perfect light settings in your images or 10 different images of a person. So what if there's only one image for each person? Our covariance estimates for the subspace may be horribly wrong, so will the recognition. Remember the Eigenfaces method had a 96% recognition rate on the AT&T Facedatabase? How many images do we actually need to get such useful estimates? Here are the Rank-1 recognition rates of the Eigenfaces and Fisherfaces method on the AT&T Facedatabase, which is a fairly easy database:



So in order to get good recognition rates you'll need at least 8(+-1) images for each person and the Fisherfaces method doesn't really help here. The above experiment is a 10-fold cross validated result carried out with the facerec framework at: https://github.com/bytefish/facerec. This is not a publication, so I won't back these figures with a deep mathematical analysis. Please have a look into for a detailed analysis of both methods, when it comes to small training datasets.

So some research concentrated on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector, but describe only local features of an object. The features you extract this way will have a low-dimensionality implicitly. A fine idea! But you'll soon observe the image representation we are given doesn't only suffer from illumination variations. Think of things like scale, translation or rotation in images - your local description has to be at least a bit robust against those things. Just like SIFT, the Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called Local Binary Patterns or sometimes referred to as LBP codes. The first LBP operator described in literature actually used a fixed 3 x 3 neighborhood just like this:



➔ Algorithmic Discription

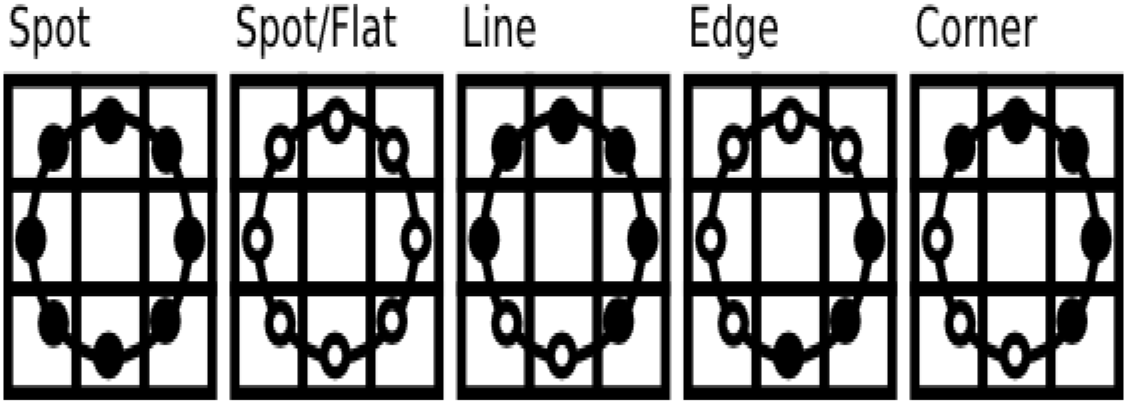
A more formal description of the LBP operator can be given as:

LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)

, with (x_c, y_c) as central pixel with intensity i_c; and i_n being the intensity of the the neighbor pixel. s is the sign function defined as:

\begin{equation}
s(x) =
\begin{cases}
1 & \text{if $x \geq 0$}\\
0 & \text{else}
\end{cases}
\end{equation}

This description enables you to capture very fine grained details in images. In fact the authors were able to compete with state of the art results for texture classification. Soon after the operator was published it was noted, that a fixed neighborhood fails to encode details differing in scale. So the operator was extended to use a variable neighborhood in [AHP04]. The idea is to align an abritrary number of neighbors on a circle with a variable radius, which enables to capture the following neighborhoods:



For a given Point (x_c,y_c) the position of the neighbor (x_p,y_p), p \in P can be calculated by:

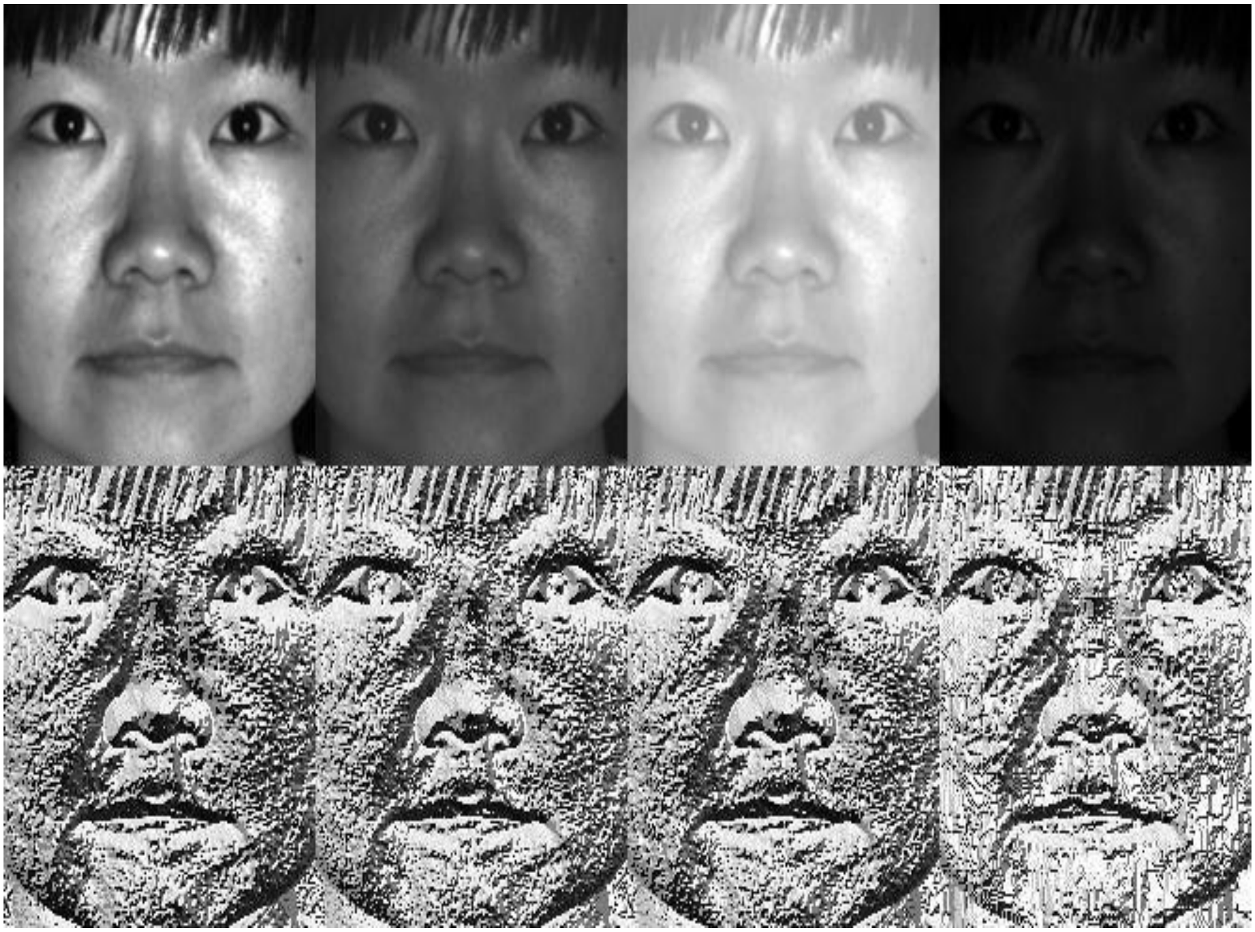
```
\begin{align*}
x_{p} & = & x_c + R \cos({\frac{2\pi p}{P}})\\
y_{p} & = & y_c - R \sin({\frac{2\pi p}{P}})
\end{align*}
```

Where R is the radius of the circle and P is the number of sample points.

The operator is an extension to the original LBP codes, so it’s sometimes called Extended LBP (also referred to as Circular LBP) . If a points coordinate on the circle doesn’t correspond to image coordinates, the point get’s interpolated. Computer science has a bunch of clever interpolation schemes, the OpenCV implementation does a bilinear interpolation:

```
\begin{align*}
f(x,y) \approx \begin{bmatrix}
    1-x & x \end{bmatrix} \begin{bmatrix}
    f(0,0) & f(0,1) \\
    f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix}
    1-y \\
    y \end{bmatrix}.
\end{align*}
```

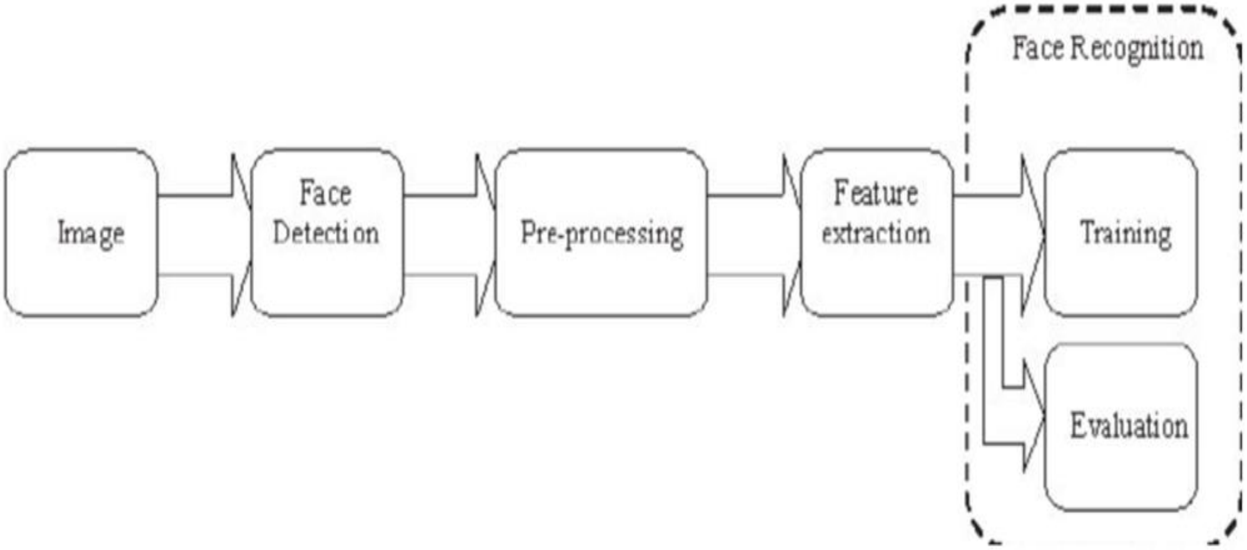
By definition the LBP operator is robust against monotonic gray scale transformations. We can easily verify this by looking at the LBP image of an artificially modified image (so you see what an LBP image looks like!):

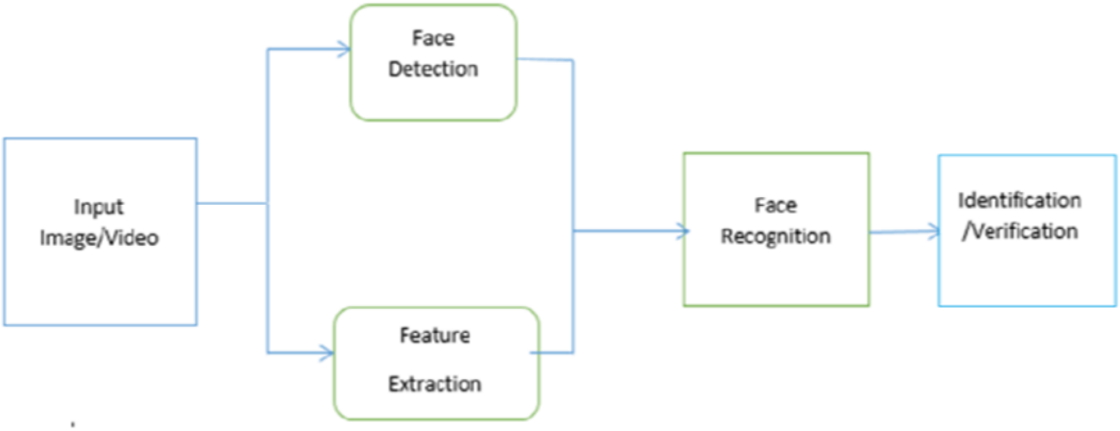


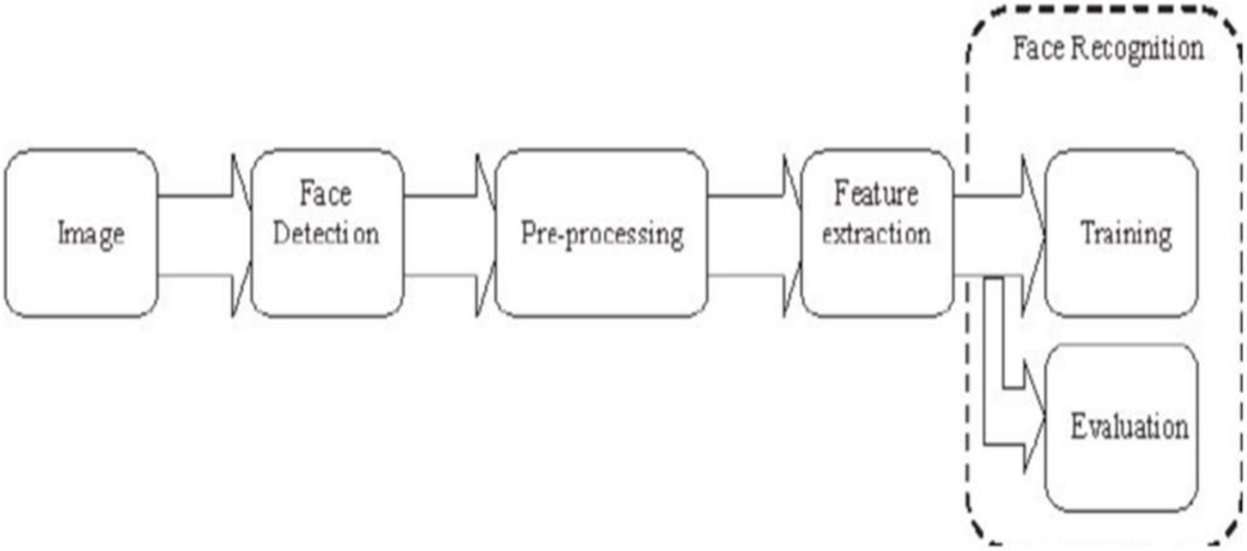
So what’s left to do is how to incorporate the spatial information in the face recognition model. The representation proposed by Ahonen et. al is to divide the LBP image into m local regions and extract a histogram from each. The spatially enhanced feature vector is then obtained by concatenating the local histograms (not merging them). These histograms are called Local Binary Patterns Histograms.

2.3 Project Design

The architecture and block diagram of the face detection and Recognization system is as follows.

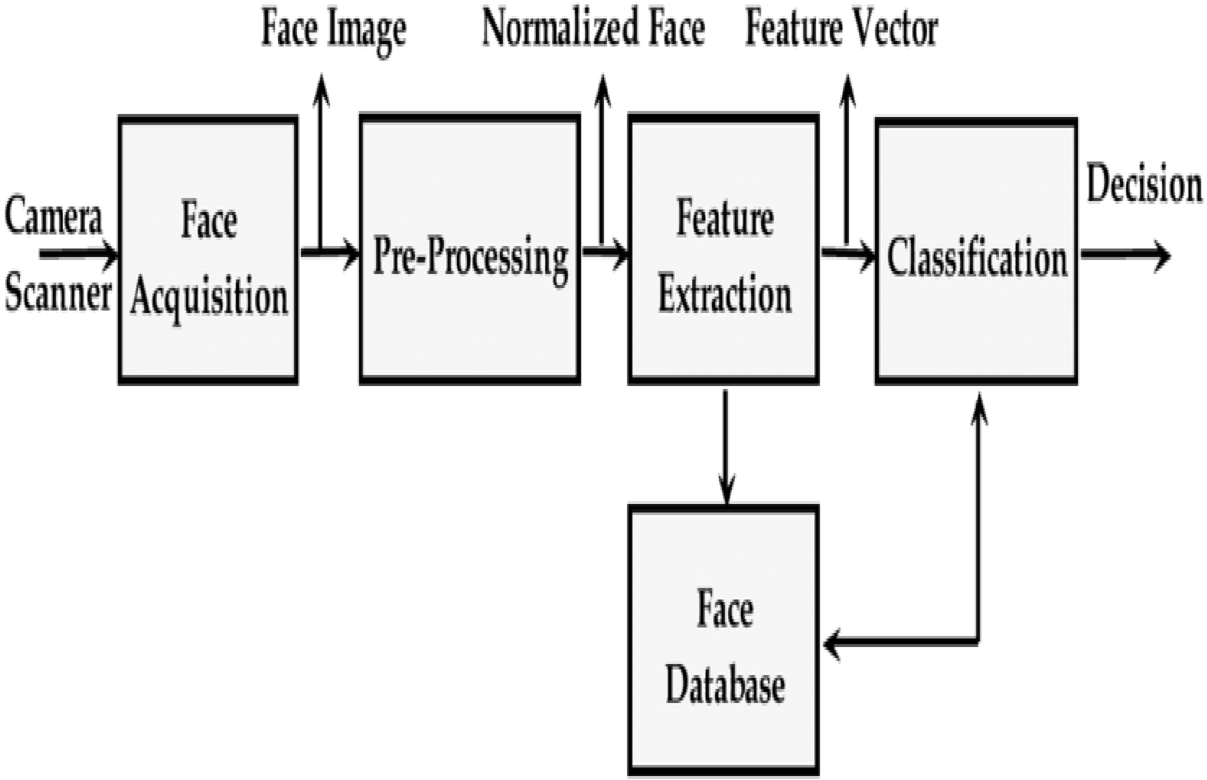





Face Recognition
Image
Face Detection
Pre-processing
Feature extraction
Training
Evaluation

CHAPTER-3 Diagram and Modules

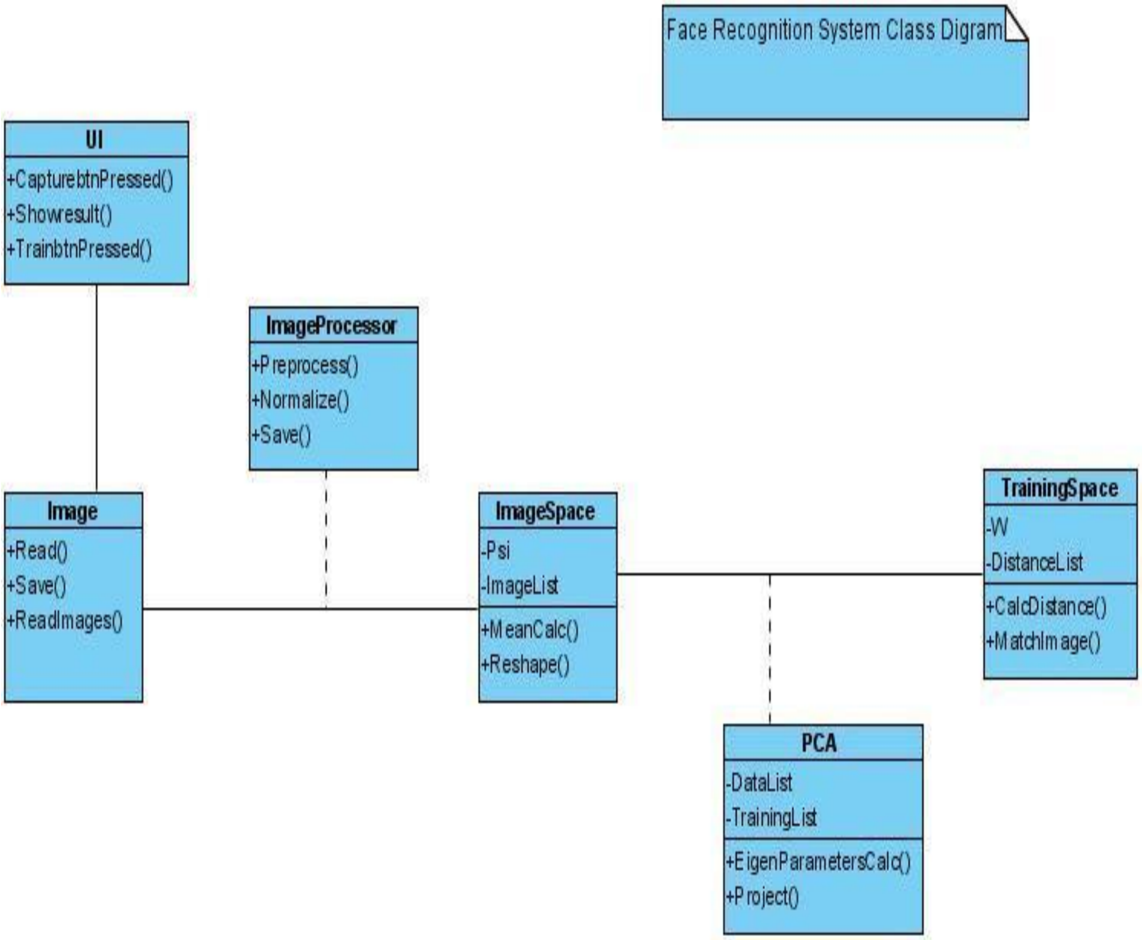
3.1 Modules of the system



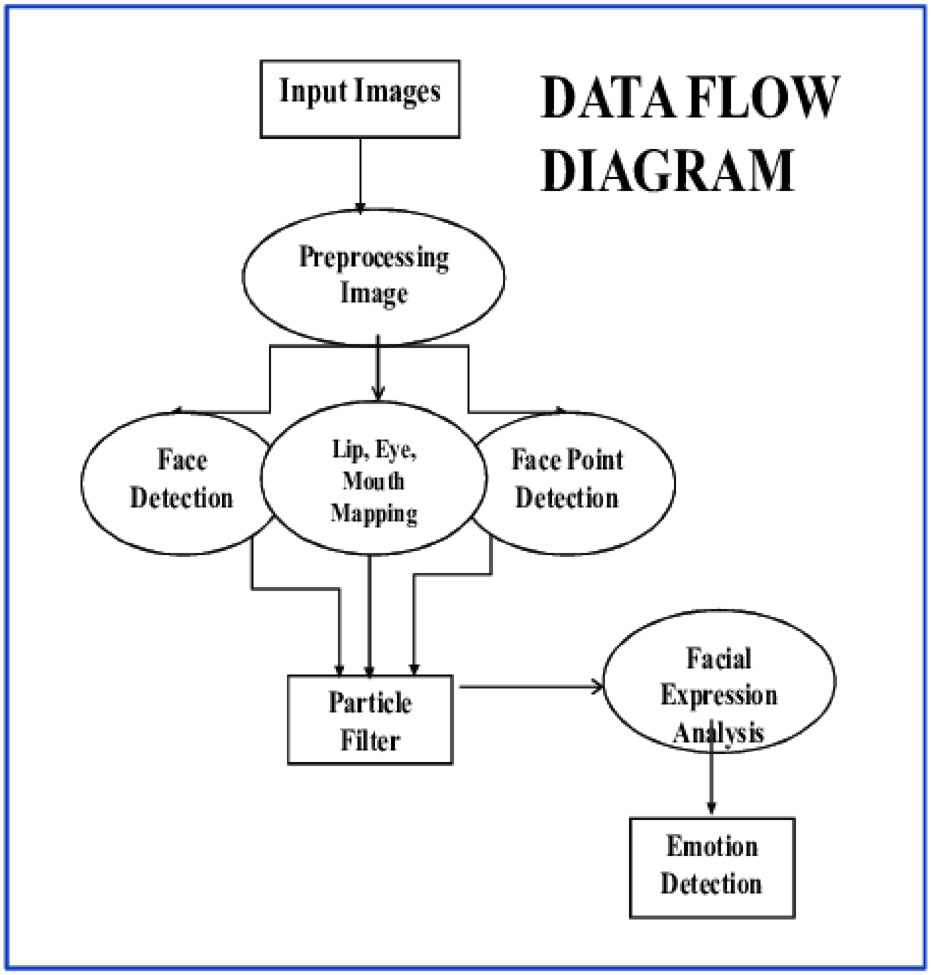
- Face Acquisition
- Pre-Processing
- Feature Extraction
- Face Database
- Classification

3.2 Diagrams

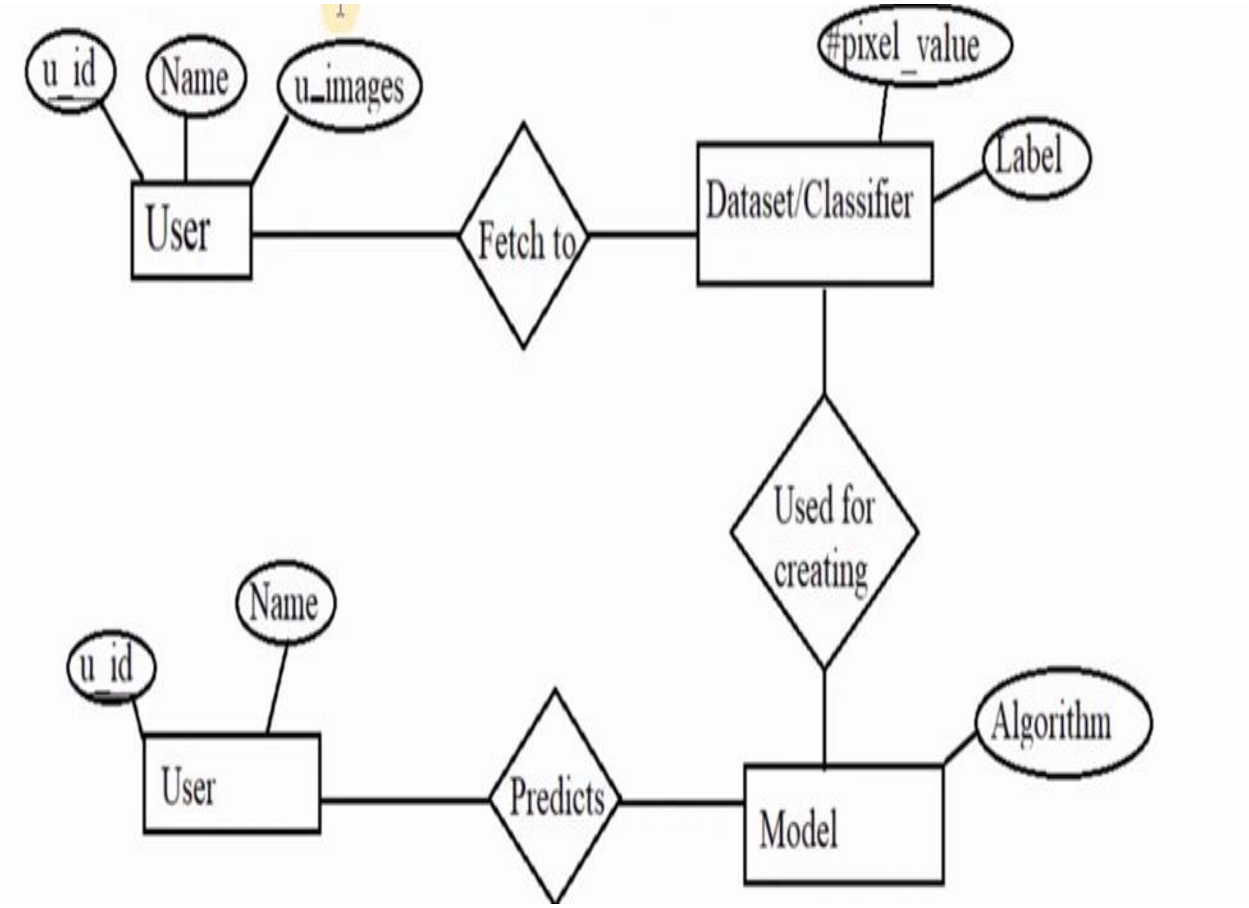
(a) Class Diagram



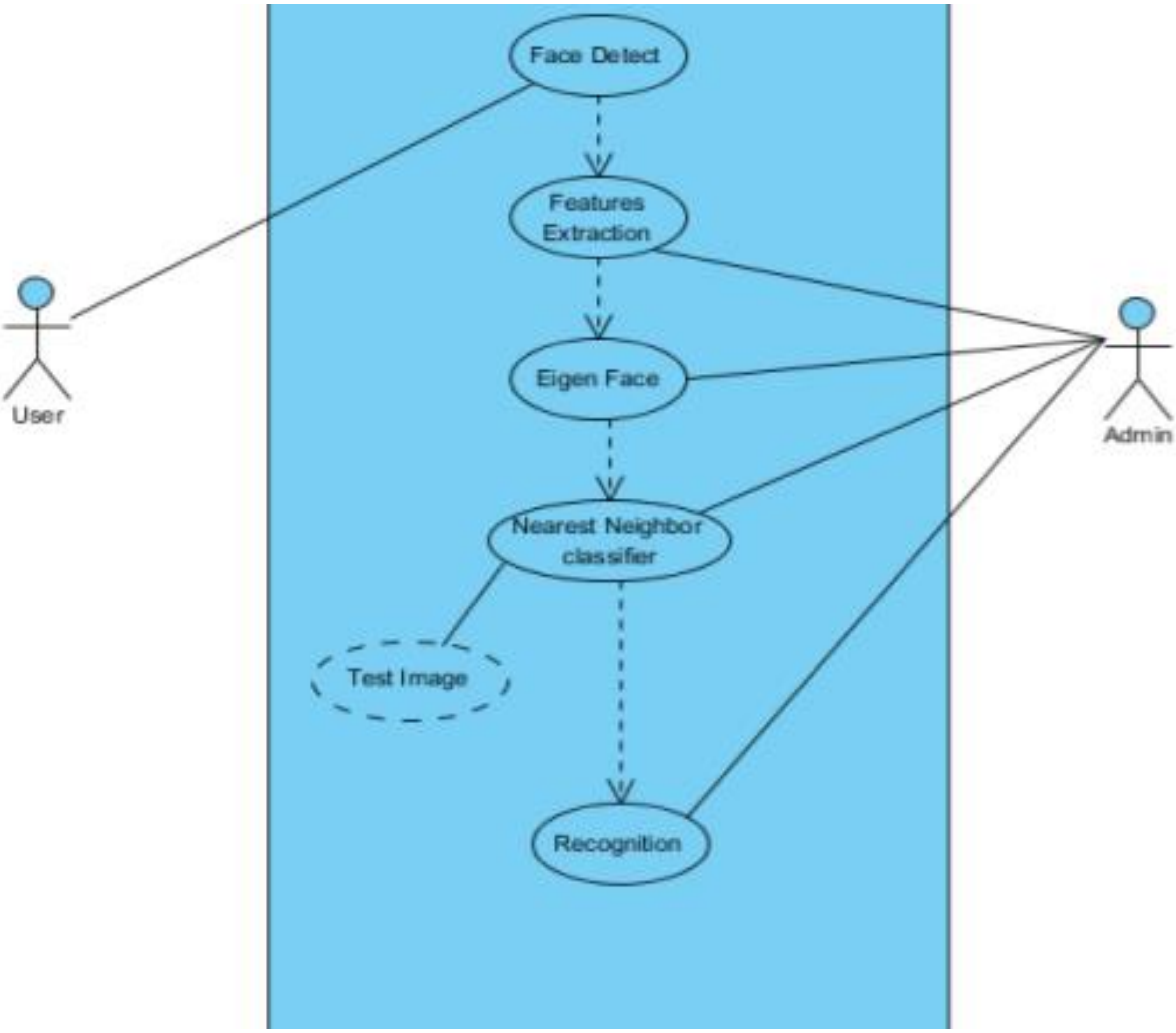
(b) Data Flow Diagram



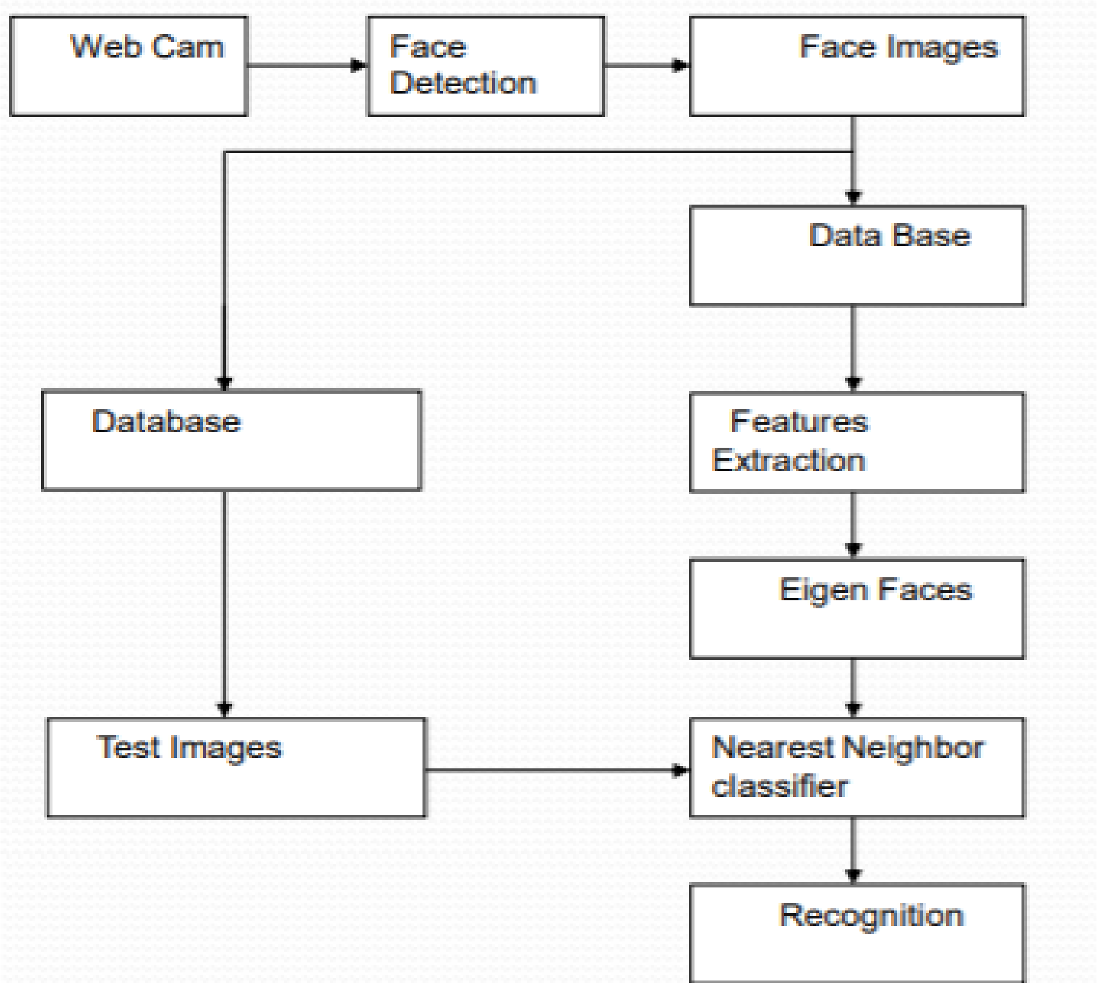
(c) E-R Diagram


u_id
Name
u_images
#pixel_value
Label
User
Fetch to
Dataset/Classifier
Used for creating
Name
u_id
Algorithm
User
Predicts
Model

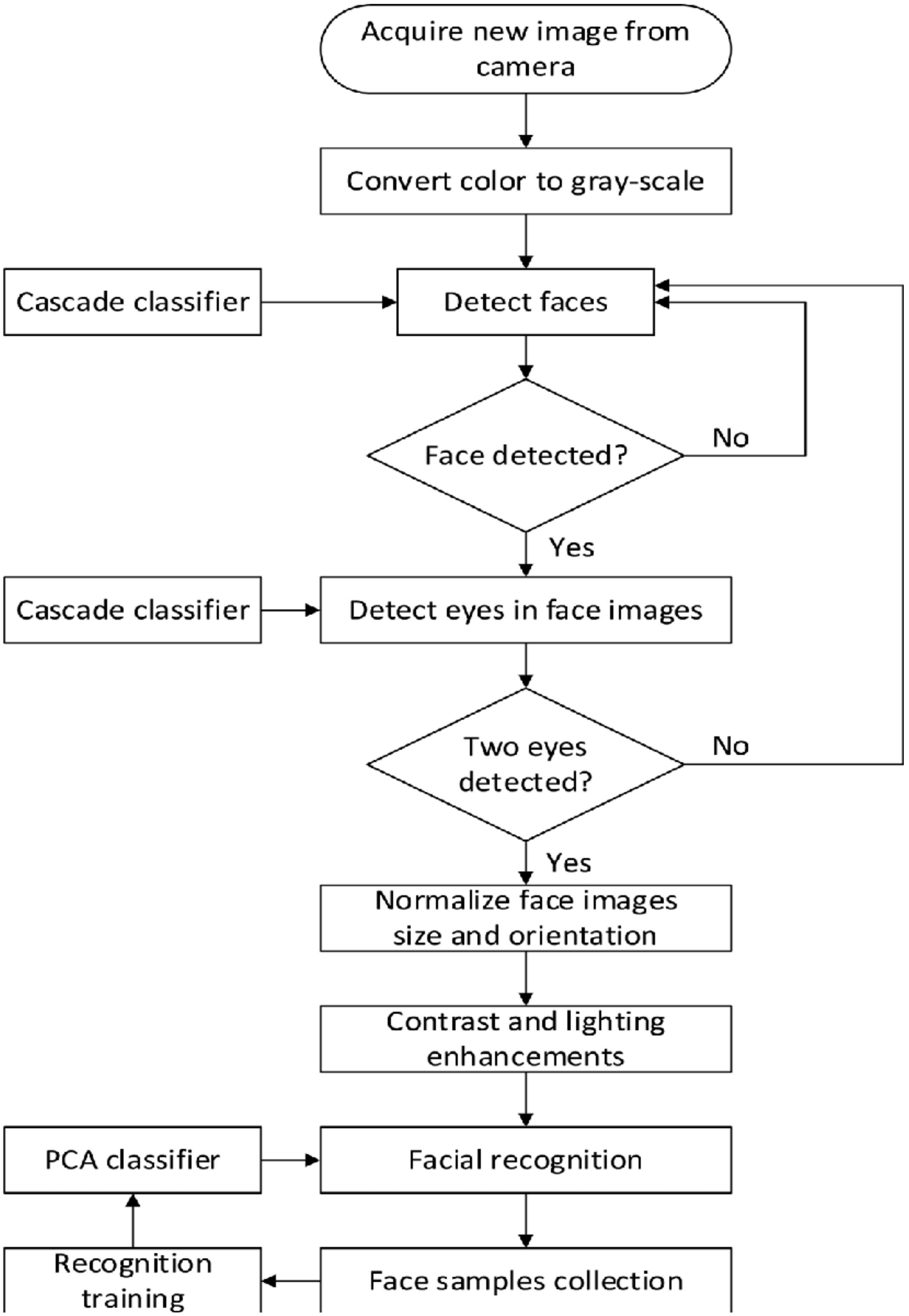
(d) Use Case Diagram



(e) Architectural Diagram



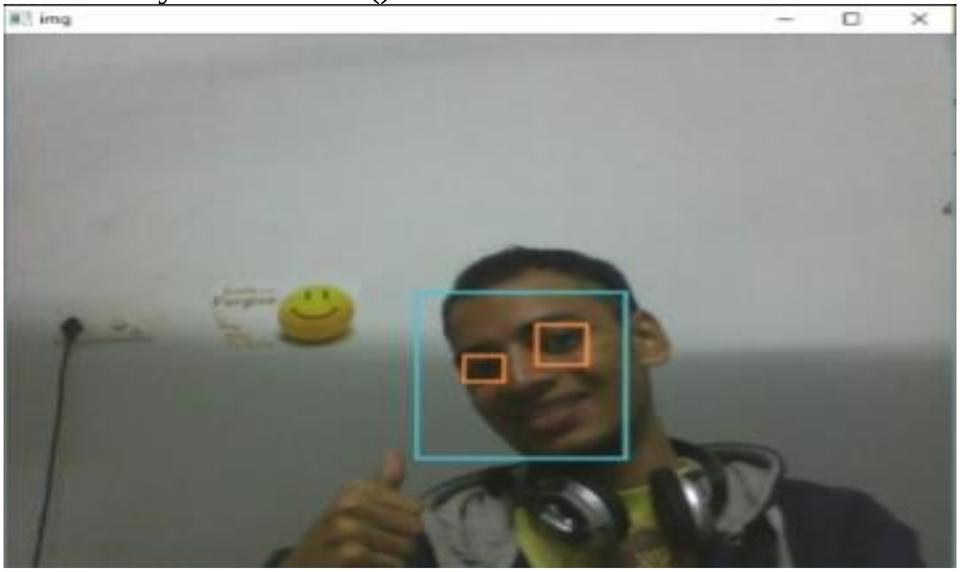
(f) Flow chart


Acquire new image from camera
Convert color to gray-scale
Cascade classifier
Detect faces
Face detected?
No
Yes
Cascade classifier
Detect eyes in face images
Two eyes detected?
No
Yes
Normalize face images size and orientation
Contrast and lighting enhancements
PCA classifier
Facial recognition
Recognition training
Face samples collection

CHAPTER-4 IMPLENTATION

Some of the code is Attached below:-

```
# OpenCV program to detect face in real time
# import libraries of python OpenCV
# where its functionality resides
import cv2

# load the required trained XML classifiers
# https://github.com/Itseez/opencv/blob/master/
# data/haarcascades/haarcascade_frontalface_default.xml
# Trained XML classifiers describes some features of some
# object we want to detect a cascade function is trained
# from a lot of positive(faces) and negative(non-faces)
# images.
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# https://github.com/Itseez/opencv/blob/master
# /data/haarcascades/haarcascade_eye.xml
# Trained XML file for detecting eyes
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

# capture frames from a camera
cap = cv2.VideoCapture(0)

# loop runs if capturing has been initialized.
while 1:

    # reads frames from a camera
    ret, img = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detects faces of different sizes in the input image
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:
        # To draw a rectangle in a face
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        # Detects eyes of different sizes in the input image
        eyes = eye_cascade.detectMultiScale(roi_gray)

        #To draw a rectangle in eyes
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,127,255),2)

    # Display an image in a window
    cv2.imshow('img',img)

    # Wait for Esc key to stop
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

# Close the window
cap.release()

# De-allocate any associated memory usage
cv2.destroyAllWindows()
```



CHAPTER- 5 RESULT

To improve recognition performance, there are many things that can be improved here, some of them fairly easy to implement. For example, color processing, eye detection, upper- body, and lower-body detection, etc. You can usually improve the face recognition accuracy by using more input images, at least 50 per person, by taking more photos of each person, particularly from different angles and lighting conditions. Based on the results we can conclude that this program works when the model or person is near to the camera. For future prediction, this paper can be used for keeping the track or the details of criminals so that their recognition could be easy. This can also be used for the betterment of human- machine interaction so that machines can understand the feelings or the emotions of the human by scanning their faces. It can also be used in advanced attendance manager so that teachers don’t have to take attendance, with the help of this software they can easily recognize the child’s face and can do the attendance.

CHAPTER-6 CONCLUSION

The computational models, which were implemented in this project, were chosen after extensive research, and the successful testing results confirm that the choices made by the researcher were reliable.The system with manual face detection and automatic face recognition did not have a recognition accuracy over 90%, due to the limited number of eigenfaces that were used for the PCA transform. This system was tested under very robust conditions in this experimental study and it is envisaged that real-world performance will be far more accurate. B The fully automated frontal view face detection system displayed virtually perfect accuracy and in the researcher's opinion further work need not be conducted in this area. The fully automated face detection and recognition system was not robust enough to achieve a high recognition accuracy. The only reason for this was the face recognition subsystem did not display even a slight degree of invariance to scale, rotation or shift errors of the segmented face image. This was one of the system requirements identified in section 2.3. However, if some sort of further processing, such as an eye detection technique, was implemented to further normalise the segmented face image, performance will increase to levels comparable to the manual face detection and recognition system. Implementing an eye detection technique would be a minor extension to the implemented system and would not require a great deal of additional research.All other implemented systems displayed commendable results and reflect well on the deformable template and Principal Component Analysis strategies.The most suitable real-world applications for face detection and recognition systems are for mugshot matching and surveillance. There are better techniques such as iris or retina recognition and face recognition using the thermal spectrum for user access and user verification applications since these need a very high degree of accuracy.The real-time automated pose invariant face detection and recognition system proposed in chapter seven would be ideal for crowd surveillance applications. If such a system were widely implemented its potential for locating and tracking suspects for law enforcement agencies is immense. The implemented fully automated face detection and recognition system (with an eye detection system) could be used for simple surveillance applications such as ATM user security, while the implemented manual face detection and automated recognition system is ideal of mugshot matching. Since controlled conditions are present when mugshots are gathered, the frontal view face recognition scheme should display a recognition accuracy far better than the results, which were obtained in this study, which was conducted under adverse conditions. Department of ECE Page 48 Furthermore, many of the test subjects did not present an expressionless, frontal view to the system. They would probably be more compliant when a 6'5" policeman is taking their mugshot! In mugshot matching applications, perfect recognition accuracy or an exact match is not a requirement. If a face recognition system can reduce the number of images that a human operator has to search through for a match from 10000 to even a 100, it would be of incredible practical use in law enforcement. The automated vision systems implemented in this thesis did not even approach the performance, nor were they as robust as a human's innate face recognition system. However, they give an insight into what the future may hold in computer vision

REFERENCE

- http://zetcode.com/python/pillow
- http://tutorial.simplecv.org/en/latest/
- https://www.programiz.com/python- programming/ method
  s/ built-in/ord
- https://www.academia.edu/people/searc h?utf8=%E2%9C
  %93&q=face+recognition+u sing+python
- https://www.analyticsvidhya.com/blog/2 019/03/opencvfunctions-computer-vision-python/
- https:// www. geeksforgeeks.org/ python- programminglanguage/
  IJESC, September 2019 23763 http://ijesc.org/
- https://www.academia.edu/Documents/in/Image_Process
  ing _ with_OpenCV
- https://www.techopedia.com/definition /32071/facial-recognition
- https://stackoverflow.com/questions/5 3926657/215 assert
  ion- failed-src-empty- in-function-cvcvtcolor