CENTRALIZED CRIMINAL DATA EXTRACTION USING FACE RECOGNITION

Project Report

Submitted by

Deepak Kumar Mandal (1501106507)

David Johnson Ekka (1501106506)

Akshit Mittal (1501106496)

Under the guidance of

Mr. Sunil Kumar Mohapatra

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING AND TECHNOLOGY, BHUBANESWAR

2019

CERTIFICATE

This is to certify that the report entitled Centralized Criminal Data Extraction using Face Recognition is submitted combinedly by Deepak Kumar Mandal (1501106507), David Johnson Ekka (1501106506), Akshit Mittal (1501106496), to the Department of Computer Science and Engineering, College of Engineering and Technology, Bhubaneswar. They have sincerely worked on the topic under our supervision and we consider it worthy of consideration for evaluation of the final year seminar under Biju Patnaik University of Technology, Rourkela, Odisha.

Date:

(Dr. Subasish Mohapatra)
HOD, CSE
CET, Bhubaneswar

(Mr. Sunil Kumar Mohapatra)
Lecturer, Dept. of CSE
CET, Bhubaneswar

ACKNOWLEDGEMENT

I am pleased to acknowledge Mr. Sunil Kumar Mohapatra Sir for her invaluable guidance during the course of this seminar. I extend my sincere thanks to Dr. Subasish Mohapatra sir, who continuously helped me throughout the process and without his guidance, the completion would have been an uphill task. They were a persevering boost which helped me tackle with the pressure and their contribution in simulating suggestions and encouragement, helped me to coordinate my seminar especially in writing of this report. Moreover, the constructive criticism guided me in best of direction to flaunt my unsurpassable skills.

DECLARATION

I certify that the work contained in this report is original and has been done myself under the general supervision of my supervisor. The work not been submitted to any other Institute for any degree or diploma. I have followed the guidelines provided by the Institute in writing the report. I have confirmed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever I have used material (data, theoretical analysis, figures, text) from the other sources, I have given due credit to them citing them in the text of the report and giving their details in the references. Whenever I have quoted written material from other sources I have put them under quotation and given due credit to the sources by citing them and giving required details in the references.

(Deepak Kumar Mandal) (David Johnson Ekka) (Akshit Mittal)

ABSTRACT

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. 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. The software requirements for this project is OpenCV software. Identifying a person with an image has been popularized through the mass media. However, it is less robust to fingerprint or retina scanning. This presentation describes the face detection and recognition minor-project. It reports the technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Python. For face detection, Haar-Cascades were used and for face recognition Local binary pattern histograms were used. The methodology is described including flow charts for each stage of the system. Next, the results are shown including plots and screen-shots followed by a discussion of encountered challenges. The presentation is concluded with the opinion on the project and possible applications.

KEYWORDS :-

Face Recognition, numpy, python , haarcascade, openCV,

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter-1 : INTRODUCTION

The following document is a report on the mini project for Robotic visual perception and autonomy. It involved building a system for face detection and face recognition using several classifiers available in the open computer vision library(OpenCV). Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analysed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision ”whose face is it ? ”, using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, a conclusion is provided on the pros and cons of each algorithm and possible implementations.

1.1 FACE RECOGNIZATION:

DIFFERENT APPROACHES OF FACE RECOGNITION:

There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

Recognition algorithms can be divided into two main approaches:

1. Geometric: Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features. (Figure 3)

2. Photometric stereo: Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006) (Figure 1)

Popular recognition algorithms include:

1. Principal Component Analysis using Eigenfaces, (PCA)

2. Linear Discriminate Analysis,

3. Elastic Bunch Graph Matching using the Fisher face algorithm,

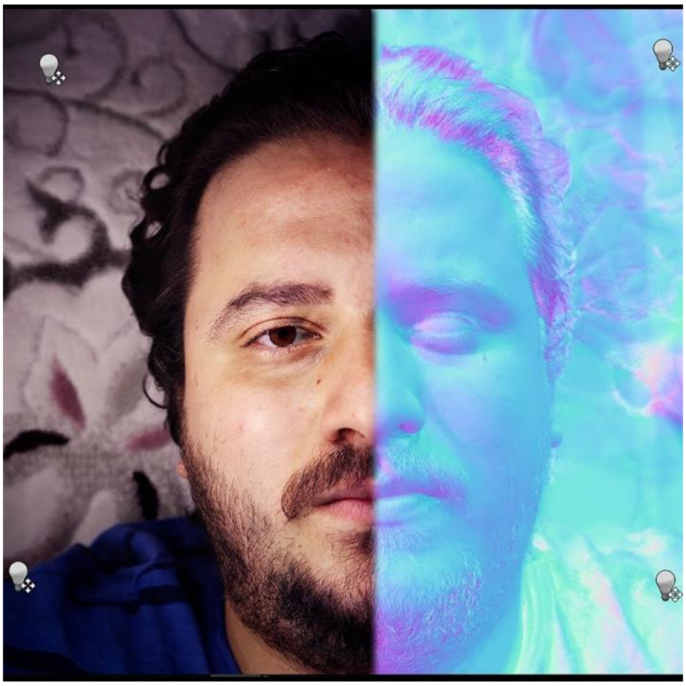

Figure 1 - Photometric Stereo Image

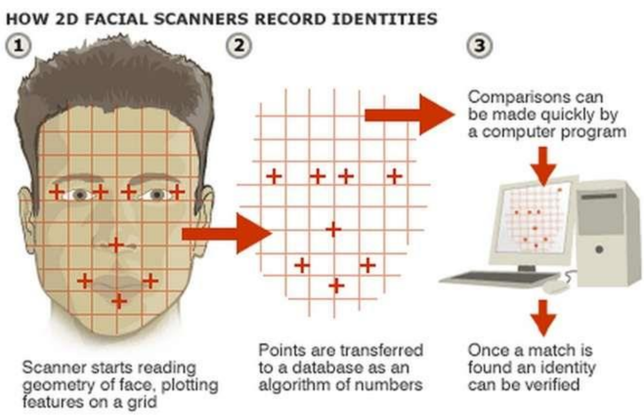

Figure 2 – Geometrical Face Recognition

1.2 FACE DETECTION:

Face detection involves separating image windows into two classes; one containing faces (tarning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin colour and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

The face detection system can be divided into the following steps:-

1. Pre-Processing: To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples that is the face images are obtained by cropping Department of ECE Page 3 images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.

2. Classification: Neural networks are implemented to classify the images as faces or nonfaces by training on these examples. We use both our implementation of the neural network and the Matlab neural network toolbox for this task. Different network configurations are experimented with to optimize the results.

3. Localization: The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has done on:- Position Scale Orientation Illumination

Chapter-2: LITERATURE SURVEY

Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection, can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically there are two types of approaches to detect facial part in the given image i.e. feature base and image base approach. Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match between training and testing images.

Table 1. Brief literature review of face recognition.

Author(s)	Title	Database	Pre-Processing	Feature extraction	Classifier	Accuracy
Yang et al. (2004)	Two-dimensional PCA: a new approach to appearance-based face representation and recognition	ORL, AR and Yale	No	Pixel + 2DPCA	kNN	ORL - 98.3% AR - 96.1% Yale - 84.24%
Zhang and Zhou (2005)	(2D)2PCA - Two-directional two-dimensional PCA for efficient face representation and recognition	ORL and FERET	Image resize 60×60	Pixel + (2D)2PCA	kNN	ORL - 90.5% FERET- 85%.
Yang et al. (2005)	Two-dimensional discriminant transformfor face recognition	ORL	No	Pixel + 2DLDA	kNN	ORL - 96.4%
Noushath et al. (2006)	(2D)2LDA: An efficient approach for face recognition	ORL	No	Pixel + (2D)2LDA	kNN	ORL - 99.75%
MageshKumar et al. (2011)	Gabor features and LDA based Face Recognition with ANN classifier	ORL	Image resize to 128×128 with Gabor filter	Coefficients of Gabor wavelet transform + PCA+ LDA	kNN and neural network	kNN - 99.75% neural network: 98.6%
Shermina (2011)	Illumination invariant face recognition using Discrete Cosine Transform and Principal Component Analysis	Yale B	Illumination normalization and correction using DCT	Pixel + PCA	Not mentioned	Yale B: 94.2%
Oliveira et al. (2011)	2D Principal Component Analysis for face and facial-expression recognition	ORL, AR, Feret, Yale and Jafe	No	Pixel + 2DPCA + Genetic algorithm	kNN and SVM	ORL - 91.5% (KNN), 93.1% (SVM); FERET - 90% (SVM); YALE - 89.2%(SVM); AR - 92.1% (SVM) Jaffe - 94% (SVM).
Le and Bui (2011)	Face recognition based on SVM and 2DPCA	FERET and ORL	Image resizing to 50×50 (FERET)	Pixel + PCA + 2DPCA	kNN and SVM	ORL - 97.3% (SVM), 96.2% (KNN); FERET - 95.1% (SVM), 90.1% (KNN).

2.1 FEATURE BASE APPROACH:

Active Shape Model Active shape models focus on complex non-rigid features like actual physical and higher level appearance of features Means that Active Shape Models (ASMs) are aimed at automatically locating landmark points that define the shape of any statistically modelled object in an image. When of facial features such as the eyes, lips, nose, mouth and eyebrows. The training stage of an ASM involves the building of a statistical facial model from a training set containing images with manually annotated landmarks. ASMs is classified into three groups i.e. snakes, PDM, Deformable templates

(A)Snakes:

The first type uses a generic active contour called snakes, first introduced by Kass et al. in 1987 Snakes are used to identify head boundaries [8,9,10,11,12]. In order to achieve the task, a snake is first initialized at the proximity around a head boundary. It then locks onto nearby edges and subsequently assume the shape of the head. The evolution of a snake is achieved by minimizing an energy function, Esnake (analogy with physical systems), denoted asEsnake = Einternal + EExternal WhereEinternal and EExternal are internal and external energy functions.Internal energy is the part that depends on the intrinsic properties of the snake and defines its natural evolution. The typical natural evolution in snakes is shrinking or expanding. The external energy counteracts the internal energy and enables the contours to deviate from the natural evolution and eventually assume the shape of nearby features—the head boundary at a state of equilibria. Two main consideration for forming snakes i.e. selection of energy terms and energy minimization. Elastic energy is used commonly as internal energy. Internal energy is vary with the distance between control points on the snake, through which we get contour an elastic-band characteristic that causes it to shrink or expand. On other side external energy relay on image features. Energy minimization process is done by optimization techniques such as the steepest gradient descent. Which needs highest computations. Huang and Chen and Lam and Yan both employ fast iteration methods by greedy algorithms. Snakes have some demerits like contour often becomes trapped onto false image features and another one is that snakes are not suitable in extracting non-convex features.

(B) Deformable Templates:

Deformable templates were then introduced by Yuille et al. to take into account the a priority of facial features and to better the performance of snakes. Locating a facial feature boundary is not an easy task because the local evidence of facial edges is difficult to organize into a sensible global entity using generic contours. The low brightness contrast around some of these features also makes the edge detection process. Yuille et al. took the concept of snakes a step further by incorporating global information of the eye to improve the reliability of the extraction process. Deformable templates approaches are developed to solve this problem. Deformation is based on local valley, edge, peak, and brightness .Other than face boundary, salient feature (eyes, nose, mouth and eyebrows) extraction is a great challenge of face recognition. E = Ev + Ee + Ep + Ei + Einternal ; where Ev , Ee , Ep , Ei , Einternal are external energy due to valley, edges, peak and image brightness and internal energy.

(C) PDM (Point Distribution Model):

Independently of computerized image analysis, and before ASMs were developed, researchers developed statistical models of shape . The idea is that once you represent shapes as vectors, you can apply standard statistical methods to them just like any other multivariate object. These models learn allowable constellations of shape points from training example sand use principal components to build what is called a Point Distribution Model. These have been used in diverse ways, for example for categorizing Iron Age broaches. Ideal Point Distribution Models can only deform in ways that are characteristic of the object. Cootes and his colleagues were seeking models which do exactly that so if a beard, say, covers the chin, the shape model can \override the image" to approximate the position of the chin under the beard. It was therefore natural (but perhaps only in retrospect) to adopt Point Distribution Models. This synthesis of ideas from image processing and statistical shape modelling led to the Active Shape Model. The first parametric statistical shape model for image analysis based on principal components of inter-landmark distances was presented by Cootes and Taylor in. On this approach, Cootes, Taylor, and their colleagues, then released a series of papers that cumulated in what we call the classical Active Shape Model.

2.2) LOW LEVEL ANALYSIS:

Based on low level visual features like color, intensity, edges, motion etc.

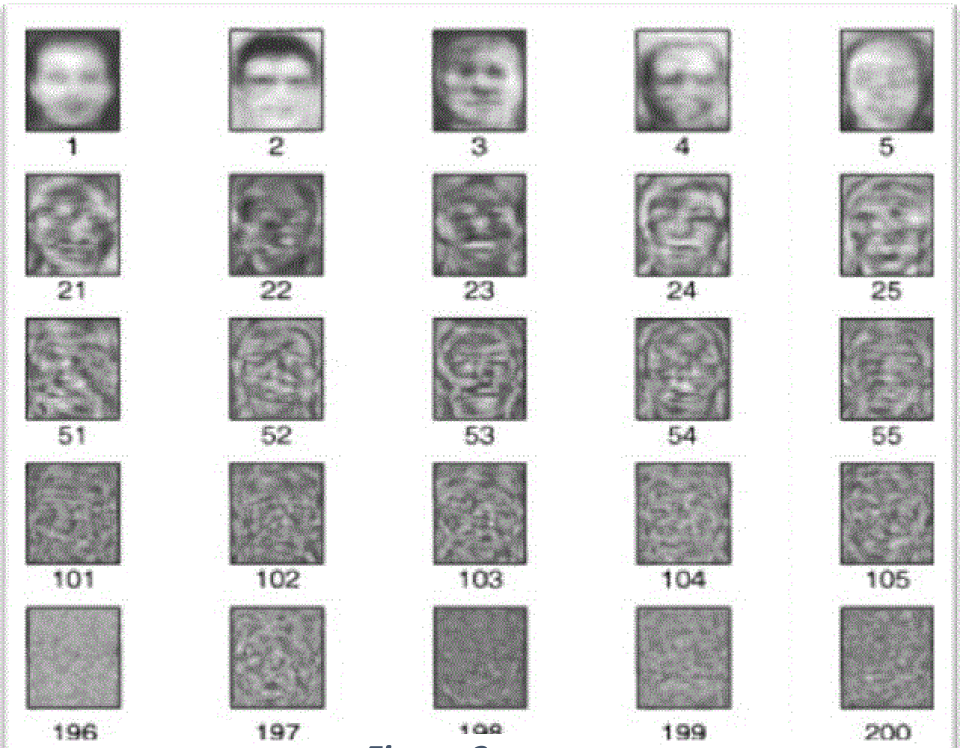

Figure 3

2.1) Skin Color Base

Color is avital feature of human faces. Using skin-color as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features. Under certain lighting conditions, color is orientation invariant. This property makes motion estimation much easier because only a translation model is needed for motion estimation [25]. Tracking human faces using color as a feature has several problems like the color representation of a face obtained by a camera is influenced by many factors (ambient light, object movement, etc.).

Majorly three different face detection algorithms are available based on RGB, YCbCr, and HIS color space models. In the implementation of the algorithms there are three main steps viz.

(1) Classify the skin region in the color space,

(2) Apply threshold to mask the skin region and

(3) Draw bounding box to extract the face image.

Crowley and Coutaz suggested simplest skin color algorithms for detecting skin pixels. The perceived human color varies as a function of the relative direction to the illumination. The pixels for skin region can be detected using a normalized color histogram, and can be normalized for changes in intensity on dividing by luminance. Converted an [R, G, B] vector is converted into an [r, g] vector of normalized color which provides a fast means of skin detection. This algorithm fails when there are some more skin region like legs, arms, etc.

Cahi and Ngan suggested skin color classification algorithm with YCbCr color space. Research found that pixels belonging to skin region having similar Cb and Cr values. So that the thresholds be chosen as [Cr1, Cr2] and [Cb1, Cb2], a pixel is classified to have skin tone if the values [Cr, Cb] fall within the thresholds. The skin color distribution gives the face portion in the color image. This algorithm is also having the constraint that the image should be having only face as the skin region. Kjeldson and Kender defined a color predicate in HSV color space to separate skin regions from background [28]. Skin color classification inHSI color space is the same as YCbCr color space but here the responsible values are hue (H) and saturation (S). Similar to above the threshold be chosen as [H1, S1] and [H2, S2], and a pixel is classified to have skin tone if the values [H,S] fall within the threshold and this distribution gives the localized face image. Similar to above two algorithm this algorithm is also having the same constraint.

2.2) Motion Base

When use of video sequence is available, motion information can be used to locate moving objects. Moving silhouettes like face and body parts can be extracted by simply thresholding accumulated frame differences [29]. Besides face regions, facial features can be located by frame differences [30, 31].

2.3) Gray Scale Base

Gray information within a face can also be treat as important features. Facial features such as eyebrows, pupils, and lips appear generally darker than their surrounding facial regions. Various recent feature extraction algorithms [32 – 34] search for local gray minima within segmented facial regions. In these algorithms, the input images are first enhanced by contrast-stretching and gray-scale morphological routines to improve the quality of local dark patches and thereby make detection easier. The extraction of dark patches is achieved by low-level gray-scale thresholding. Based method and consist three levels. Yang and huang presented new approach i.e. face gray scale behaviour in pyramid (mosaic) images. This system utilizes hierarchical Face location consist three levels. Higher two level based on mosaic images at different resolution. In the lower level, edge detection method is proposed. Moreover, this algorithm gives fine response in complex background where size of the face is unknown.

2.4) Edge Base

Face detection based on edges was introduced by Sakai et al. [36]. This work was based on analysing line drawings of the faces from photographs, aiming to locate facial features. Than later Craw et al. proposed a hierarchical framework based on Sakai et al.'work to trace a human head outline. Then after remarkable works were carried out by many researchers in this specific area. Method suggested by Anila and Devarajan was very simple and fast. They proposed frame work which consist three steps i.e. initially the images are enhanced by applying median filter for noise removal and histogram equalization for contrast adjustment. In the second step the edge images constructed from the enhanced image by applying sober operator. Then a novel edge tracking algorithm is applied to extract the sub windows from the enhanced image based on edges. Further they used Back propagation Neural Network (BPN) algorithm to classify the sub-window as either face or non-face.

3) FEATURE ANALYSIS

These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. These methods are designed mainly for face localization.

3.1) Feature Searching

3.1.1) Viola Jones Method

Paul Viola and Michael Jones presented an approach for object detection which minimizes computation time while achieving high detection accuracy. Paul Viola and Michael Jones proposed a fast and robust method for face detection which is 15 times quicker than any technique at the time of release with 95% accuracy at around 17 fps. The technique relies on the use of simple Haar-like features that are evaluated quickly through the use of a new image representation. Based on the concept of an ―Integral Image‖ it generates a large set of features and uses the boosting algorithm AdaBoost to reduce the overcomplete set and the introduction of a degenerative tree of the boosted classifiers provides for robust and fast interferences. The detector is applied in a scanning fashion and used on gray-scale images, the scanned window that is applied can also be scaled, as well as the features evaluated.

3.1.2) Gabor Feature Method

Sharif et al. proposed an Elastic Bunch Graph Map (EBGM) algorithm that successfully implements face detection using Gabor filters. The proposed system applies 40 different Gabor filters on an image. As a result of which 40 images with different angles and orientation are received. Next, maximum intensity points in each filtered image are calculated and mark them as fiducial points. The system reduces these points in accordance to distance between them. The next step is calculating the distances between the reduced points using distance formula. At last, the distances are compared with database. If match occurs, it means that the faces in the image are detected. Equation of Gabor filter is shown below gives the orientation,

$$\psi_{u,v}(z) = \frac{\| k_{u,v} \|^2}{\sigma^2} e^{\left(\frac{\|k_{u,v}\|^2 \|z\|^2}{2\sigma^2} \right)} \left[e^{i\vec{k}_{u,v} z} - e^{-\frac{\sigma^2}{2}} \right]$$

Where

$$\phi_u = \frac{u\pi}{8}, \quad \phi_u \in [0, \pi)$$ gives the frequency,

Figure 4

3.2) Constellation Method

All methods discussed so far are able to track faces but still some issue like locating faces of various poses in complex background is truly difficult. To reduce this difficulty investigator, form a group of facial features in face-like constellations using more robust modelling approaches such as statistical analysis. Various types of face constellations have been proposed by Burl et al. [41]. They establish use of statistical shape theory on the features detected from a multiscale Gaussian derivative filter. Huang et al. also apply a Gaussian filter for pre-processing in a framework based on image feature analysis.

B. IMAGE BASE APPROACH

1) Neural Network

Neural networks gaining much more attention in many pattern recognition problems, such as OCR, object recognition, and autonomous robot driving. Since face detection can be treated as a two class pattern recognition problem, various neural network algorithms have been proposed. The advantage of using neural networks for face detection is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one demerit is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance. In early days most hierarchical neural network was proposed by Agui et al. [43]. The first stage having two parallel subnetworks in which the inputs are filtered intensity values from an original image. The inputs to the second stage network consist of the outputs from the sub networks and extracted feature values. An output at the second stage shows the presence of a face in the input region. Propp and Samal developed one of the earliest neural networks for face detection [44]. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

Feraud and Bernier presented a detection method using auto associative neural networks [45], [46], [47]. The idea is based on which shows an auto associative network with five layers is able to perform a nonlinear principal component analysis. One auto associative network is used to detect frontal-view faces and another one is used to detect faces turned up to 60 degrees to the left and right of the frontal view. After that Lin et al. presented a face detection system using probabilistic decision-based neural network (PDBNN) [49]. The architecture of PDBNN is similar to a radial basis function (RBF) network with modified learning rules and probabilistic interpretation.

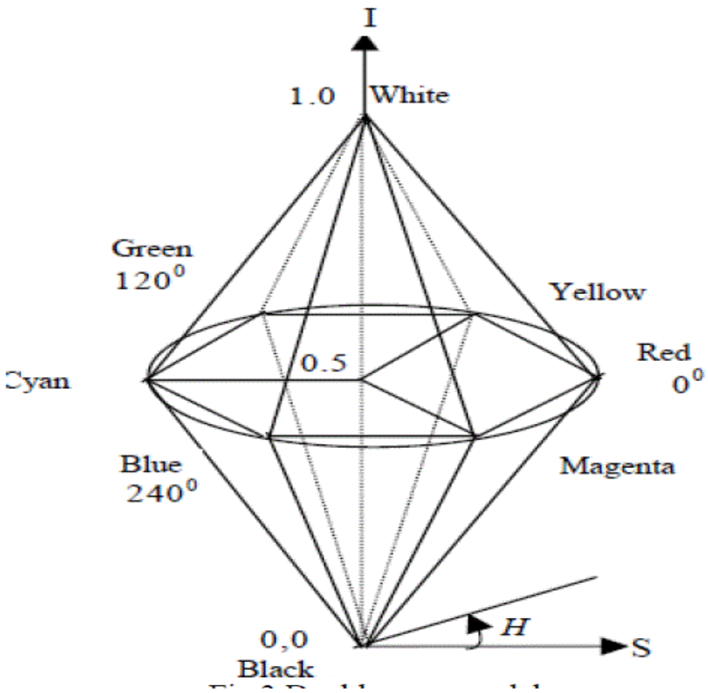

Figure 5

2) Linear Sub Space Method

2.1) Eigen faces Method

An early example of employing eigenvectors in face recognition was done by Kohonen in which a simple neural network is demonstrated to perform face recognition for aligned and normalized face images. Kirby and Sirovich suggested that images of faces can be linearly encoded using a modest number of basis images[51]. The idea is arguably proposed first by Pearson in 1901 and then by Hotelling in 1933 [53].Given a collection of n by m pixel training images represented as a vector of size m X n, basis vectors spanning an optimal subspace are determined such that the mean square error between the projection of the training images onto this subspace and the original images is minimized. They call the set of optimal basis vectors Eigen pictures since these are simply the eigenvectors of the covariance matrix computed from the vectorized face images in the training set. Experiments with a set of 100 images show that a face image of 91 X 50 pixels can be effectively encoded using only50 Eigen pictures, while retaining a reasonable likeness (i.e., capturing 95 percent of the variance).

3) Statistical Approach

3.1) Support Vector Machine (SVM)

SVMs were first introduced Osuna et al. for face detection. SVMs work as a new paradigm to train polynomial function, neural networks, or radial basis function (RBF) classifiers. SVMs works on induction principle, called structural risk minimization, which targets to minimize an upper bound on the expected generalization error. An SVM classifier is a linear classifier where the separating hyper plane is chosen to minimize the expected classification error of the unseen test patterns. IN [54], Ozonate al. developed an efficient method to train an SVMf or large scale problems ,and applied it to face detection. Based on two test sets of 10,000,000 test patterns of 19 X 19 pixels, their system has slightly lower error rates and runs approximately30 times faster than the system by Sung and Poggio [55]. SVMs have also been used to detect faces and pedestrians in the wavelet domain.

PROBLEM STATEMENT

In order to locate a human face, the system needs to capture an image using a camera and a frame-grabber to process the image, search the image for important features and then use these features to determine the location of the face. For detecting face there are various algorithms and methods including skin colour based, haar like features, adaboost and cascade classifier Colour is an important feature of human faces. Using skin-colour as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features The input of a face recognition system is always an image or video stream. The output is an identification or verification of the subject or subjects that appear in the image or video. Some approaches define a face recognition system as a three step process (Figure 1). From this point of view, the Face Detection and Feature Extraction phases could run simultaneously

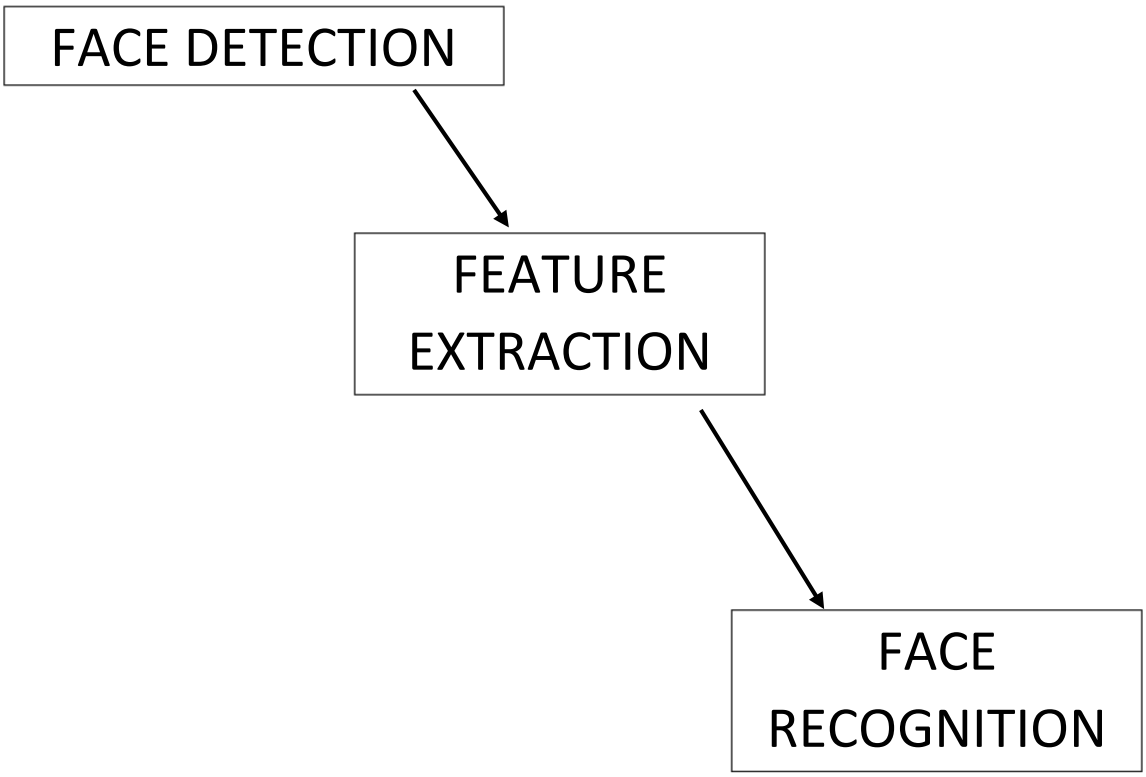

Figure 6

Chapter-3: IMPLEMENTATION

Architecture Model

To develop the artificial intelligence systems, scientists are usually based on the natural behavior of human beings. Such is the case of the automatic face recognition that seeks to rival or surpass the amazing capabilities of the human visual system. The latter so that it can identify a person from his face, he needs to have previously associated and memorized the appearance of her face with his identity. This process of learning is easily achieved in humans in a spontaneous and scalable manner. It must be integrated in an artificial face recognition system and carried out in a supervised manner. The learning or enrollment process is the first module to be executed offline in a face recognition system. While the second module that addresses online, is that of recognition (authentication or identification). The first module allows extracting feature vectors (signatures) from the images of the base reference faces (Gallery); the database of face images of people assumed to be known by the system. These are the extracted feature vectors are learned or stored for use in the classification phase. While the recognition module extracts the feature vector of the query face image which is subjected to the input of the system for the first time. In these two modules, one must browse the processing steps illustrated in Figure 1 and described below. Face detection and localization: the captured image can contain both the face of the person to recognize and possibly a background. Hence, first of all one must detect the presence or absence of the face in the captured image. If the image contains a face, its location is localized to extract it. This step is performed in two modules (recruitment and recognition). Feature extraction: this is the key step of the process because the performance of the whole system depends on it. In this step also known as indexing or modeling, is extracted from the detected face image a characteristic vector (signature) that is sufficiently representative of a given face and which models the much more precise than the raw image departure. This new representation of the face must have both the uniqueness property for each person and the property of discrimination between different people. Learning: in this step, stores the extracted feature vectors of individuals known to the system. During this learning process, each vector stored in the database is associated with a certain personal identity such as name, personal identification number which characterizes the user. This reference database is centralized to a central server or distributed on a smart card only has to recognize the person based on the intended application. Classification and decision: in these two steps, the system must declare the identity of the person who appears before them without any a priori knowledge about it. To accomplish this task, we must affect the extracted feature of his face to a class from those learned. Each class is associated with an identity. These two steps are executed only in the recognition module.

Face Detection: Look at the picture and find a face in it.

Data Gathering: Extract unique characteristics of face that it can use to differentiate him from another person, like eyes, mouth, nose, etc.

Data Comparison: Despite variations in light or expression, it will compare those unique features to all the features of all the people you know.

Face Recognition: It will determine the known face.

Our mind's Face Recognition Process is like, the more you meet the person, the more data you will collect about him, and the quicker your mind will be able to recognize him. Or, at least you should. Whether or not you are good with names is another story. Here is when it gets better: Our human brains are wired to do all these things automatically. In fact, we are very good at detecting faces almost everywhere: Computers aren't able, yet, to do this automatically, so we need to *teach them* how to do it step-by-step. However, you probably assumed that it's incredibly difficult to code your computer to recognize faces, right?

3.1. THEORY OF OPENCV FACE RECOGNIZERS

Thanks to OpenCV, coding facial recognition is now easier than ever. There are three easy steps to computer coding facial recognition, which are similar to the steps that our brains use for recognizing faces. These steps are:

1. Data Gathering: Gather face data (face images in this case) of the persons you want to identify.
2. Train the Recognizer: Feed that face data and respective names of each face to the recognizer so that it can learn.
3. Recognition: Feed new faces of that people and see if the face recognizer you just trained recognizes them.

OpenCV has three built-in face recognizers and thanks to its clean coding, you can use any of them just by changing a single line of code. Here are the names of those face recognizers and their OpenCV calls:

EigenFaces – cv2.face.createEigenFaceRecognizer()

FisherFaces – cv2.face.createFisherFaceRecognizer()

Local Binary Patterns Histograms
(LBPH) – cv2.face.createLBPHFaceRecognizer()

3.1.1 EIGENFACES FACE RECOGNIZER

This algorithm considers the fact that not all parts of a face are equally important or useful for face recognition. Indeed, when you look at someone, you recognize that person by his distinct features, like the eyes, nose, cheeks or forehead; and how they vary respect to each other.

In that sense, you are focusing on the areas of maximum change. For example, from the eyes to the nose there is a significant change, and same applies from the nose to the mouth. When you look at multiple faces, you compare them by looking at these areas, because by catching the maximum variation among faces, they help you differentiate one face from the other. In this way, is how EigenFaces recognizer works. It looks at all the training images of all the people as a whole and tries to extract the components which are relevant and useful and discards the rest. These important features are called principal components.

Note: We will use the terms: principal components, variance, areas of high change and useful features indistinctly as they all mean the same.Below is an image showing the variance extracted from a list of faces

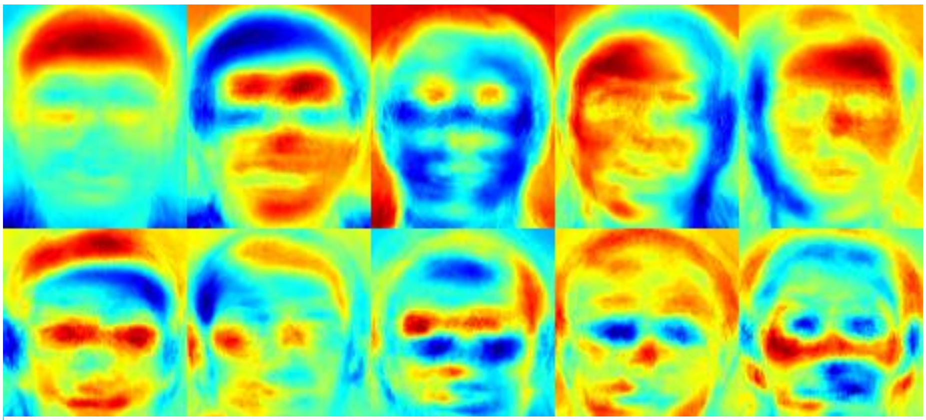

Figure 7 - EigenFaces Face Recognizer

You can see that the useful features represent faces which receive the name of Eigen Faces. So, EigenFaces recognizer trains itself by extracting principal components, but it also keeps a record of which ones belong to which person. Thus, whenever you introduce a new image to the algorithm, it repeats the same process as follows:

1. Extract the principal components from the new picture.
2. Compare those features with the list of elements stored during training.
3. Find the ones with the best match.
4. Return the 'person' label associated with that best match component.

In simple words, it's a game of matching.

However, one thing to note in above image is that EigenFaces algorithm also considers illumination as an important feature. In consequence, lights and shadows are picked up by EigenFaces, which classifies them as representing a 'face.'

Face recognition picks up on human things, dominated by shapes and shadows: two eyes, a nose, a mouth.

3.1.2 FISHERFACES FACE RECOGNIZER

This algorithm is an improved version of the last one. As we just saw, EigenFaces looks at all the training faces of all the people at once and finds principal components from all of them combined. By doing that, it doesn't focus on the features that discriminate one individual from another.

Consider the lighting changes in following images:

Figure 8

Since EigenFaces also finds illumination as a useful component, it will find this variation very relevant for face recognition and may discard the features of the other people's faces, considering them less useful. In the end, the variance that EigenFaces has extracted represents just one individual's facial features.

To fix this issue , We can do it by tuning EigenFaces so that it extracts useful features from the faces of each person separately instead of extracting them from all the faces combined. In this way, even if one person has high illumination changes, it will not affect the other people's features extraction process. Precisely, FisherFaces face recognizer algorithm extracts principal components that differentiate one person from the others. In that sense, an individual's components do not dominate (become more useful) over the others.

Below is an image of principal components using FisherFaces algorithm.

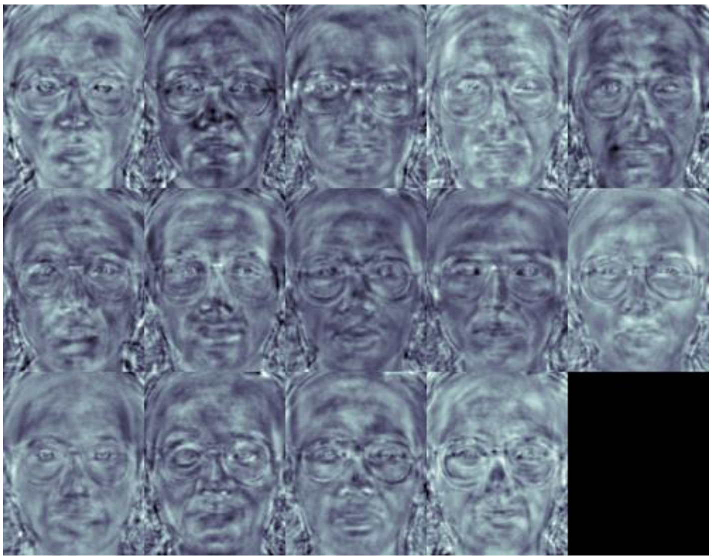

Figure 9 - FisherFaces Face Recognizer Principal Components.

One thing to note here is that FisherFaces only prevents features of one person from becoming dominant, but it still considers illumination changes as a useful feature. We know that light variation is not a useful feature to extract as it is not part of the actual face.

3.2 LOCAL BINARY PATTERNS HISTOGRAMS (LBPH) FACE RECOGNIZER

We know that Eigenfaces and Fisherfaces are both affected by light and, in real life, we can't guarantee perfect light conditions. LBPH face recognizer is an improvement to overcome this drawback.The idea with LBPH is not to look at the image as a whole, but instead, try to find its local structure by comparing each pixel to the neighboring pixels.

THE LBPH FACE RECOGNIZER PROCESS

Take a 3×3 window and move it across one image. At each move (each local part of the picture), compare the pixel at the center, with its surrounding pixels. Denote the neighbors with intensity value less than or equal to the center pixel by 1 and the rest by 0.

After you read these 0/1 values under the 3×3 window in a clockwise order, you will have a binary pattern like 11100011 that is local to a particular area of the picture. When you finish doing this on the whole image, you will have a list of local binary patterns.

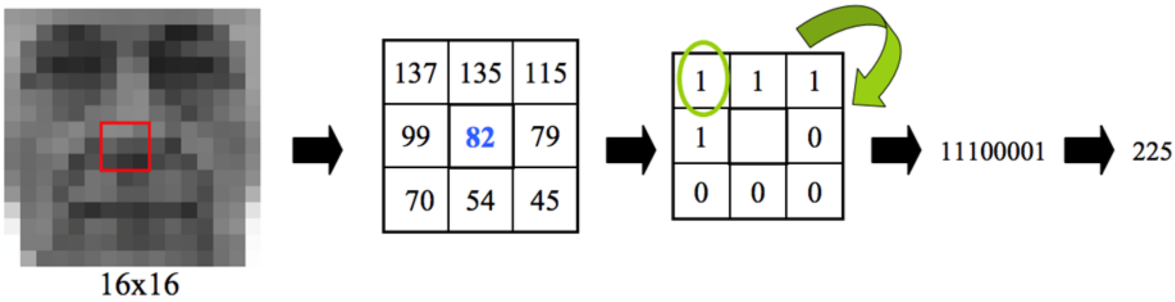

Figure 10 - LBP conversion to binary. (Source: López & Ruiz; Local Binary Patterns applied to Face Detection and Recognition.)

Now, after you get a list of local binary patterns, you convert each one into a decimal number using binary to decimal conversion (as shown in above image)

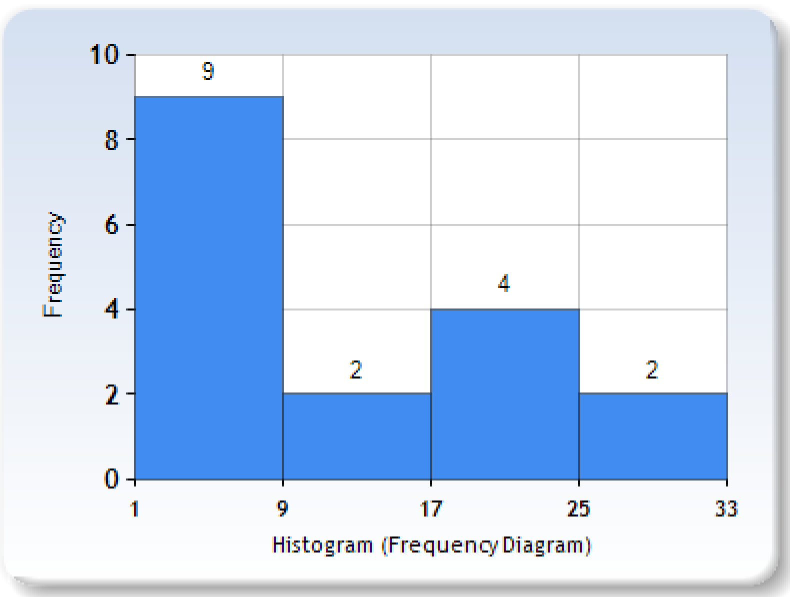

Figure 11

and then you make a histogram of all of those decimal values. A sample histogram looks like this:

Histogram Sample.

In the end, you will have one histogram for each face in the training data set. That means that if there were 100 images in the training data set then LBPH will extract 100 histograms after training and store them for later recognition. Remember, the algorithm also keeps track of which histogram belongs to which person.

Later during recognition, the process is as follows:

1. Feed a new image to the recognizer for face recognition.
2. The recognizer generates a histogram for that new picture.
3. It then compares that histogram with the histograms it already has.

Finally, it finds the best match and returns the person label associated with that best match.

Below is a group of faces and their respective local binary patterns images. You can see that the LBP faces are not affected by changes in light conditions:

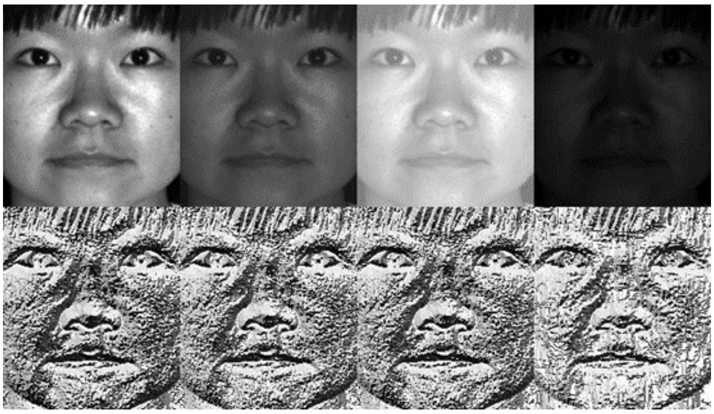

Figure 12 - LBPH Face Recognizer Principal Components.

3.3 CODING FACE RECOGNITION USING PYTHON AND OPENCV

The Face Recognition process is divided into three steps:

1) Prepare Training Data: Read training images for each person/subject along with their labels, detect faces from each image and assign each detected face an integer label of the person it belongs.
2) Train Face Recognizer: Train OpenCV's LBPH recognizer by feeding it the data we prepared in step 1.
3) Prediction: Introduce some test images to face recognizer and see if it predicts them correctly.

To detect faces, I will use the code from my previous article on face detection.

Before we start the actual coding, we need to install the Code Dependencies and import the Required Modules:

CODE DEPENDENCIES

Install the following dependencies:

1. OpenCV 3.2.0
2. Python v3.5
3. NumPy that makes computing in Python easy. It contains a powerful implementation of N-dimensional arrays which we will use for feeding data as input to OpenCV functions.

REQUIRED MODULES

Import the following modules:A

1. cv2: This is the OpenCV module for Python used for face detection and face recognition.
2. os: We will use this Python module to read our training directories and file names.
3. numpy: This module converts Python lists to numpy arrays as OpenCV face recognizer needs them for the face recognition process.

3.1 PREPARE TRAINING DATA.

The more images used in training, the better.Being thorough with this principle is important because it is the only way for training a face recognizer so it can learn the different ‘faces’ of the same person; for example: with glasses, without glasses, laughing, sad, happy, crying, with a beard, without a beard, etc.

So, our training data consists of total two people with 12 images of each one. All training data is inside the folder:training-data.

This folder contains one subfolder for every individual, named with the format: sLabel (e.g. s1, s2) where the label is the integer assigned to that person. For example, the subfolder called s1 means that it contains images for person 1.

With that in mind, the directory structure tree for training data is as follows:

```
training-data
|-------------- s1
|               |-- 1.jpg
|               |-- ...
|               |-- 12.jpg
|-------------- s2
|               |-- 1.jpg
|               |-- ...
|               |-- 12.jpg
```

Figure 13

On the other hand, The foldertest-data contains images that we will use to test our face recognition program after we have trained it successfully.

Considering that the OpenCV face recognizer only accepts labels as integers, we need to define a mapping between integer tags and the person’s actual name.

So, below I am defining the mapping of a person’s integer labels and their respective names.

Note: As we have not assigned label 0 to anyone, the mapping for tag 0 is empty:

DATA PREPARATION FOR FACE RECOGNITION.

Well, to know which face belongs to which person, OpenCV face recognizer accepts information in a particular format. In fact, it receives two vectors:

- One is the faces of all the people.
- The second is the integer labels for each face.

For example, if we had two individuals and two images for each one:

PERSON-1	PERSON-2
img1	img1
img2	img2

Figure 14

Then, the data preparation step will produce following face and label vectors:

FACES	LABELS
person1_img1_face	1
person1_img2_face	1
person2_img1_face	2
person2_img2_face	2

Figure 15

In detail, we can further divide this step into the following sub-steps:

1. Read all the sub folders names provided in the folder training-data. In this tutorial; we have folder names:s1, s2.
2. Extract label number. Remember that all the sub folders containing images of a person following the format:sLabel where Label is an integer representing each person. So for example, folder name: s1 means that the person has label 1, s2 means the person's label is 2, and so on. We will assign the integer extracted in this step to every face detected in the next one.
3. Read all the images of the person, and apply face detection to each one.

3.4 TRAIN FACE RECOGNIZER.

OpenCV comes equipped with three face recognizers.

1. EigenFaces: cv2.face.createEigenFaceRecognizer()
2. FisherFaces: cv2.face.createFisherFaceRecognizer()
3. Local Binary Patterns Histogram (LBPH): cv2.face.LBPHFisherFaceRecognizer()

We are going to use now LBPH recognizer this time and see if my theory about the ghost of Elvis stealing my followers is right. It doesn't matter which of the OpenCV's face recognition programs you use because the code will remain the same. You just have to change one line, which is the face recognizer initialization line given below.

Now that we have initialized our face recognizer and we also have prepared our training data, it's time to train. We will do that by calling the methodtrain(faces-vector, labels-vector) of face recognizer. Instead of passing vectorlabels directly to face recognizer, we are first converting it to numpy array? The reason is that OpenCV expects labels vector to be a numpyarray.

3.5 PREDICTION

This is where we get to see if our algorithm is recognizing our individual faces or not. We're going to take one test image of each person, use face detection and then pass those faces to our trained face recognizer. Then we find out if our face recognition is successful. Below are some utility functions that we will use for drawing bounding box (rectangle) around the face and putting the person's name near the face bounding box.

The first functiondraw_rectangle draws a rectangle on the image based on given coordinates. It uses OpenCV's built in functioncv2.rectangle(img, topLeftPoint, bottomRightPoint, rgbColor, lineWidth) to do so.The second functiondraw_text uses OpenCV's built in functioncv2.putText(img, text, startPoint, font, fontSize, rgbColor, lineWidth) to draw text on the image. Now that we have the drawing functions, we just need to call the face recognizer's predict(face) method to test our face recognizer on test images

Chapter-4:-FUTURE WORK

The proposed work and research has been motivated by the need for a person identification system which is non-invasive and accurate. The face and ear automatic recognition system works well for image segmentation by employing both color and depth information. While the system achieved results from research and experiments, from these also came new ideas and areas which could be explored in some future research and work. Feature extraction is crucial for any recognition algorithm and system. A remarkable change has been noticed in the recognition rate using preprocessing and different feature extraction. Specifically in cropping an image before it is run through a recognition system, there is still much work to be done in this area. It would be interesting to explore new techniques of preprocessing that would lead to the optimal recognition rates. In this case, face and ear biometrics has been used, but perhaps there are other metrics that can be combined that will lead to a more robust system. It would be interesting to see what kind of results could be achieved in a system such as this with other metrics. The system provides the template security for the biometric fuzzy vault database, but still it can be investigate for the large scale operation and integrating another biometric with the face and ear feature set. The joint efforts of the whole face recognition research community have made few applications of real-world face recognition achievable, but there are still several challenges to address and opportunities to explore for designing mature face systems that can work in highly challenging environments and with images typically found within the social media environment. Hence, significant new research is required to address the aforementioned technical challenges of face recognition. To address these issues, either invariant features should be extracted to describe face image or devolve modern machine learning algorithms for learning robust representation of the face. Besides, the following points can be investigated as research directions to enhance face recognition performance to deal with emerging new and more demanding scenarios as well as working in outdoor conditions. For the illumination problem, the human face can be treated as a combination of a sequence of small and flat facets. For each facet, the effect of the illumination can be modelled using effective mathematical representation or by extracting facial features which are invariant to illumination variations. In this context, there are some trials that attempted to construct a generative 3D face model that can be used to render face images with different poses and under varying lighting conditions [177].Despite much attention to face illumination preprocessing, there are few systemic comparative studies on existing approaches that present fascinating insights and conclusions in how to design better illumination preprocessing methods [55]. Moreover, adding specifically designed preprocessing method before feature extraction can effectively increase the performance of face recognition system in a case of sever variations as well as developing methods that utilize color information instead of using grey images. Moreover, most of the existing works that address the problem of matching faces across changes in pose and illumination cannot be applied when the gallery and probe images are of different resolutions[178]. There is a need for approaches that can match an LR probe image with a high-resolution gallery via using, for example, super-resolution techniques to construct a higher resolution image from the probe image and then perform matching. Face recognition, similarly to other domains, for example, optical character recognition, has been known to benefit from the combination of multiple sources of information. Such information sources may include analysis of facial skin texture, shape of various shape parts, ratios of distances in the face, facial symmetry or lack thereof etc. Wolf et al. [82] show that combining several descriptors from the same LBP family boosts recognition performance. This suggests that, even though the development of new descriptors is an experimental science which is guided by best practices more than by solid theory, there is room for the introduction of new face encoding methods. Recent studies [179] have shown that fractional differential operator, which is a generalization of the integer-order differentiation, can strengthen high-frequency marginal information and extract more detailed information contained in objects where grey scales intensively change and textural information in those areas. Fractional differential operator has not been investigated for face representation/recognition. In fact, face images contain rich textural information and edge information, thus, fractional differential operator may extract facial visual features of local regions such as eyes, nose and mouth etc. Soft biometric traits embedded in a face such as demographic information (e.g. gender and ethnicity) and facial marks(e.g. scars, moles and freckles) are ancillary information and are not fully distinctive by themselves in face recognition tasks [180].This information can be explicitly combined with face matching score to improve the overall face recognition accuracy. Moreover, in visual surveillance application, where a face image is occluded or is captured in off-frontal pose, soft biometric traits can provide even more valuable information for face matching [181]. On the other hand, facial marks can be useful to differentiate identical twins whose global facial appearances are very similar. We believe that using soft biometric traits can improve face image matching and recognition performance. Currently, the available face recognition approaches that use the2D information of face images, suffer from low reliability because of their sensitivity to illumination conditions, facial expressions and changes in pose [182]. The inadequate performance of these approaches should come as no surprise as these 2D-basedalgorithms ignore the fact that the human face is naturally a complex 3D object characterized by distinguishing features such as eyes, mouth and nose and by skin pigmentation; consequently, it needs to be described by a 3D model taking into account geometrical information of the 3D face [183]. In this context, the3D structure of the human face intuitively provides high discriminatory information and is less sensitive to variations in environmental conditions like illumination or viewpoint. Besides, historically 3D face recognition has been criticized for lack of real-world 3D sensory cameras; this issue may be resolved in few years with inexpensive 3D sensors. Therefore we argue for further research towards using new modalities captured by 3D sensors such as infrared and depth images to enhance the performance of face recognition systems. A further path of future research would be the implementation of the algorithms on parallel processing units such as GPU. One drawback of several current face recognition methods (e.g. Gabor)is their computational complexity, which keeps these methods from being widely used in real-world commercial products. Without any doubt, if one could perform classification in frame rate, such methods would be put into more practical use. Moreover, one can enhance matching by combining top-down and bottom-up information concurrently. For example, the detection and segmentation performed in the top layers can guide the bottom layer matching process and the bottom layer can enhance the segmentation in the upper layer. Exploring the effectiveness of the current methods on a large-scale unconstrained real-world face recognition problem based on images taken from the Facebook social networking web site, Flickr, YouTube etc. is necessary to design successful systems. As well as, more study needs to be carried out on utilizing the face recognition in conjunction with other biometrics such as iris, fingerprint, speech and ear recognition in order to enhance the recognition performance of these approaches (recent research[184]). Finally, there is much work to be done in order to realize methods that reflect how humans recognize faces and optimally make use of the temporal evolution of the appearance of the face for recognition.6 Conclusions Face recognition is a challenging problem in the field of computer vision, which has received a great deal of attention over the past years because of its several applications in various domains. Although research efforts have been conducted vigorously in this area, achieving mature face recognition systems for operating under constrained conditions, they are far from achieving the ideal of being able to perform adequately in all various situations that are commonly encountered by applications in the real world. This paper on face recognition serves as a reference point towards an objective evaluation of the community's progress on face recognition research and to better address the challenges of face recognition in the real-world scenarios. In this paper, we have reviewed the current achievements in face recognition and discussed several challenges and key factors that can significantly affect performance of the face recognition systems. Moreover, this paper aims to exploit the use of face recognition technology in other scientific and daily life applications. Also several possible research directions for improving the performance of the state-of-the-art face recognition systems are suggested as future directions. Finally, this paper concludes by arguing that the next step in the evolution of face recognition algorithms will require radical and courageous steps forward in terms of the face representations/descriptors, as well as the learning algorithms used.

CONCLUSIONS

The report presented a framework for the automatic detection and recognition of individuals using face and ear as input modes. The research has been undertaken and motivated due to the great demand for such a system by security services, agencies, investigative services etc. The system detector worked very well and it overcame certain obstacles by using some innovative techniques as explained in the thesis. The system uses a multimodal approaches to improve the recognition rates, and used face and ear as input. The two biometrics face and ear combines to significantly improve the recognition rates as was shown in the experimental section over the same metrics used in an all methods. 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 LBHP face Recognition. 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. 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. However, if some sort of further processing, such as an eye detection technique, was implemented to further normalize 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 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.

REFERENCES

[1] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," Proc. IEEE, vol. 83, pp. 705–740, 2015.

[2] H. Wechsler, P. Phillips, V. Bruce, F. Soulie, and T. Huang, "Face Recognition: From Theory to Applications", Springer-Verlag, 2014.

[3] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face recognition: A literature survey," CVL Technical Report, University of Maryland, 2012.

[4] S. Gong, S.J. McKenna, and A. Psarrou, Dynamic Vision: from Images to Face Recognition, Imperial College Press and World Scientific Publishing, 2010.

[5] http://www.cognitec-systems.de/Contact/contact.html

[6] http://www.eyematic.com

[7] http://www.viisage.com

[8] http://www.identix.com

[9] http://www.biometricgroup.com

[10] R. Hietmeyer, "Biometric identification promises fast and secure processing of airline passengers," The Int'l Civil Aviation Organization Journal, vol. 55, no. 9, pp. 10–11, 2012.

[11] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E Tabassi, and J.M. Bone, "FRVT 2002: Overview and summary," March 2003

[12] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Towards automatic simulation of ageing effects on face images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pp. 442–455, 2002.

[13] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[14] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, Mar. 2000.

[15] M.S. Bartlett, H.M. Lades, and T.J. Sejnowski, "Independent component representations for face recognition," in Proceedings of SPIE, 1998, vol. 3299, pp. 528–539.

[16] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,” IEEE Trans. Pattern Analysis and Machine Intelligence,vol. 19, no. 7, pp. 711–720, Jul. 2003.

[17] D. L. Swets and J. Weng, “Using discriminant eigenfeatures for image retrieval,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 831–836, 2006.

[18] M. Kirby and L. Sirovich, “Application of the Karhunen-Lo´eve procedure for the characterization of human faces,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1 pp. 103–108, Jan. 1990.

[19] ORL face database, <http://www.uk.research.att.com/facedatabase.html>.

[20] A. Pentland, B. Moghaddam, and T. Starner, “View-based and modular eigenspaces for face recognition,” in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 1994, pp. 84–91.

[21] B. Moghaddam, “Principal manifolds and probabilistic subspaces for visual recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 6, pp. 780–788, Feb. 2002.

[22] A. Hyvarinen, “Survey on independent component analysis,” Neural Computing Surveys, vol. 2, pp. 94–128, 1999.

[23] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, “Face recognition by independent component analysis,” IEEE Trans. Neural Networks, vol. 13, no. 6, pp. 1450–1464, 2002.

[24] A. Hyvarinen, “Fast and robust fixed-point algorithms for independent component analysis,” IEEE Trans. Neural Networks, vol. 10, no. 3, pp. 626–634, 1999.

[25] B. Scholkopf, A. Smola, and K. Muller, “Nonlinaer component analysis as a kernel eigenvalue problem,” Neural Computation, vol. 10, no. 5, pp. 1299–1319, 1998.

[26] Ming-Hsuan Yang, “Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods,” in Proc. IEEE International Conference on Automatic Face and Gesture Recognition, Washington D. C., May 2002, pp. 215–220.

[27] J.B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, pp. 2319–2323, 2000.

[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol. 290, pp. 2323–2326, 2000.

[29] Ming-Hsuan Yang, "Face recognition using extended isomap," in Proc. IEEE International Conference on Image Processing, Rochester, New York, Sep. 2002, vol. 2, pp. 117–120.

[30] A.M. Martinez and A.C. Kak, "PCA versus LDA," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[31] T. Vetter and T. Poggio, "Face recognition by elastic bunch graph matching," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 733–742, 1997.

[32] W. Zhao and R. Chellappa, "SFS based view synthesis for robust face recognition," in Proc. IEEE International Conference on Automatic Face and Gesture Recognition, 2000, pp. 285–292.

[33] T. Kanade, Picture Processing by Computer Complex and Recognition of Human Faces, PhD thesis, Kyoto University, 1973.

[34] L. Wiskott, J.M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no.7, pp. 775–779, 1997.

[35] M. Lades, J. C. Vorbruggen, J. Buhmann, Jorg Lange, C. Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," IEEE Trans. Computers, vol. 42, no. 3, pp. 300–310, Jan. 1993.

[36] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," in Proc. European Conference on Computer Vision, 1998, vol. 2, pp. 484–498.

[37] G.J. Edwards, T.F. Cootes, and C.J. Taylor, "Face recognition using active appearance models," in Proc. European Conference on Computer Vision, 1998, vol. 2, pp. 581–695.

[38] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[39] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in Proc. ACM SIGGRAPH, Mar. 1999, pp. 187–194.

[40] Volker Blanz, Sami Romdhani, and Thomas Vetter, "Face identification across different poses and illuminations with a 3D morphable model," in Proc. IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 202–207.

[41] P. Penev and J. Atick, "Local feature analysis: a general statistical theory for object representation," Network: Computation in Neural Systems, vol. 7, pp. 477–500, 1996.

[42] Ferdinando Samaria and Andy Harter, "Parameterisation of a stochastic model for human face identification," in Proc. 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL, Dec. 1994.

[43] Ara V. Nefian and Monson H. Hayes, "Face recognition using an embedded HMM," in Proc. International Conference On Audio- And Video-Based Biometric Person Authentication, 1999, pp. 19–24.

[44] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1090–1104, 2000.

[45]http://www.frvt.org

[47]http://whitechapel.media.mit.edu/pub/images

[48] http://images.ee.umist.ac.uk/danny/database.html

[49] http://www.ri.cmu.edu/projects/project 418.html

[50] http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb