A Project report on

Face Detection and Recognition For Acquiring Person's Data And Attendance Monitoring

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

M. VINITHA (17P31A0494)
Y. NOOKARAJU (17P31A04B9)
I. CHATURYA (17P31A0482)
M. V. SURYA VAMSI (17P31A0495)

Under the Esteemed Guidance of
Mr. S. DURGA PRASAD, M. Tech
Assistant Professor, ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY
(*Permanently Affiliated to JNTUK, Approved by AICTE, Accredited by NAAC)*
SURAMPALEM, ADB ROAD, E.G. DIST, A.P-533437

ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUK, Kakinada)

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that Project report entitled, "**Face Detection and Recognition For Acquiring Person's Data And Attendance Monitoring**" is being submitted by M. Vinitha (17P31A0494) ,Y. Nooka Raju (17P31A04B9), I. Chaturya (17P31A0482), M.V.Surya Vamsi(17P31A0495) in the partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING** from **ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY** during the academic period of 2017-2021.

Project Guide

Mr. S. Durga Prasad, M. Tech

Assistant Professor

Dept of ECE

Head of the Department

Dr. R.V.V. KRISHNA, M. Tech, Ph.D

Professor,

Department of ECE.

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

We are very grateful to our guide Mr. S. DURGA PRASAD, M. Tech Assistant Professor who laid the time program for the successful completion of this project. He initiated, challenged our thoughts and extended timely suggestions for which we are deeply indebted to him. We are grateful to him for his comments and insights in the preparation of this project.

We express our heartfelt gratitude and thanks to Dr. R.V.V. KRISHNA, M. Tech ,Ph.D. Professor, H.O.D of ELECTRONICS AND COMMUNICATION ENGINEERING for his technical support and valuable suggestions.

We express our sincere thanks to Dr. T.K. RAMAKRISNA RAO, M.Tech, Ph.D. Principal of Aditya College of Engineering & Technology, for his enriching thoughts and profound knowledge, which brought out project work to its completion.

We express our sincere thanks to our project co-ordinator Mr. G. SATTI BABU, M. Tech, (Ph.D.) for providing necessary support in the project.

We thank sincerely and profusely to all ECE TEACHING & NON-TEACHING STAFF MEMBERS of our college, for their valuable guidance.

We also like to express our gratitude to our college management and all those who have indirectly helped us in the successful completions of our project.

With Sincere Regards,

M. VINITHA (17P31A0494)
Y. NOOKARAJU (17P31A0B9)
I. CHATURYA (17P31A0482)
M. V. SURYA VAMSI (17P31A0495)

ABSTRACT

The face is the one of the easiest ways to distinguish the individual identity of each other. In this digital era, face detection and recognition plays a vital role in almost every sector. 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 planes, 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. It is used for security, authentication identification, and has got many more advantages. Despite of having low accuracy when compared to iris recognition and fingerprint recognition, it is being widely used due to its contactless and non-invasive process. Furthermore, face detection and recognition can be used for acquiring biodata and attendance marking in schools, colleges, offices, etc. This system aims to build an organization biodata and attendance system which uses the concept of face detection and recognition as existing manual database and attendance system is the time consuming and cumbersome to maintain. This system consists of five phases-database creation, face detection, face recognition, acquiring biodata, attendance updation. In this project, face databases will be created to pump data into the recognizer algorithm.It can use your existing hardware infrastructure, existing cameras and image capture devices will work with no problems. The software requirements of this project are vscode, Jupiter notebook and the programming languages required are python3, MySQL, Pandas and SQL database.

Keywords: face detection, python3, MySQL, Webcam

CONTENTS

List of contents **PAGE NO.s**

LIST OF FIGURES

CHAPTER-1

1.1 INTRODUCTION

Face recognition is the task of identifying an already detected object as a known or unknown face.Often the problem of face recognition is confused with the problem of face detection .Face Recognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

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 several 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 normal (Zhao and Chellappa, 2006) (Figure 2)

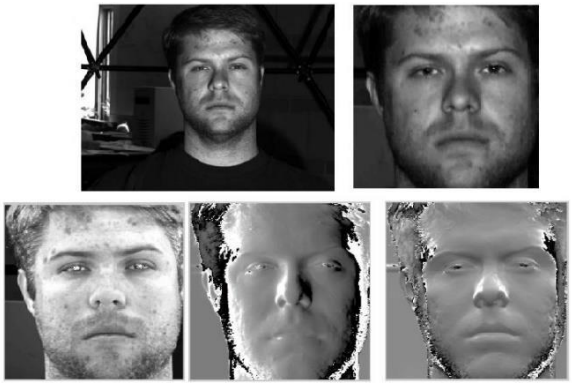


Figure 1 - Photometric Stereo Image

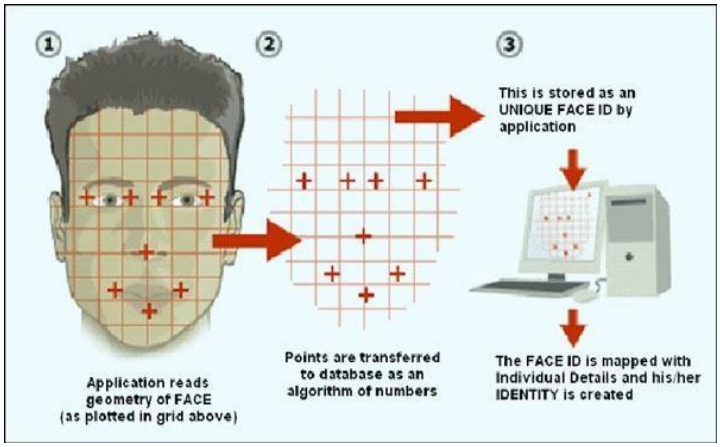


Figure 2 – Geometric Facial Recognition

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 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 matplotlib 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: -

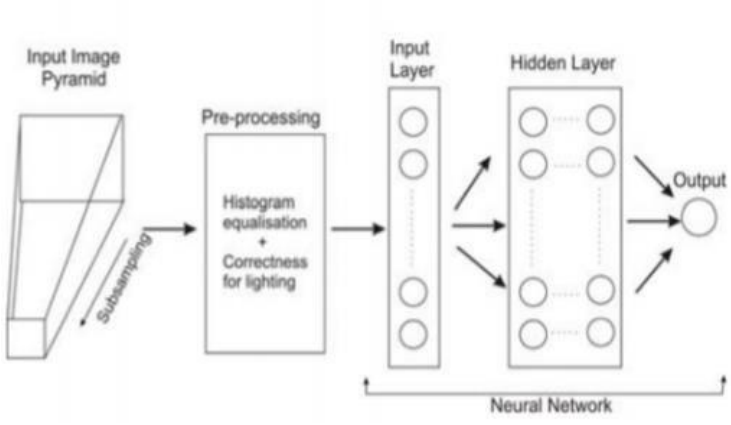


Fig 3.position scale orientation Illumination

1.3 ATTENDANCE MONITORING:

After successfully recognizing human faces, the recognized face images will be fetched into our attendance system to mark attendance of individuals.

1.3.1 Attendance Database

In our attendance database, we will extract each recognized face images to mark the attendance of individuals, using date and time educed from recognized face images as shown in Table.

ID	NAME	TIME IN
1.	I. Chaturya	19:34:10
2.	M. Vinitha	19:34:56
3.	M. Surya Vamsi	19:36:03
4.	Y. Nookaraju	19:37:30
	22-06-2021	

Table : Attendance system

Attendance of students in a large classroom is hard to be handled by the traditional system, as it is time-consuming and has a high probability of error during the process of inputting data into the computer.The proposed method using PCA to resolved the problems such as lightning of the images, noise from the camera, and the direction of the student faces. The attendance of the student was updated to the Excel

sheet after student's face has been recognized.A face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals are mailed to the respective faculty.

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 moregeneralcase 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.

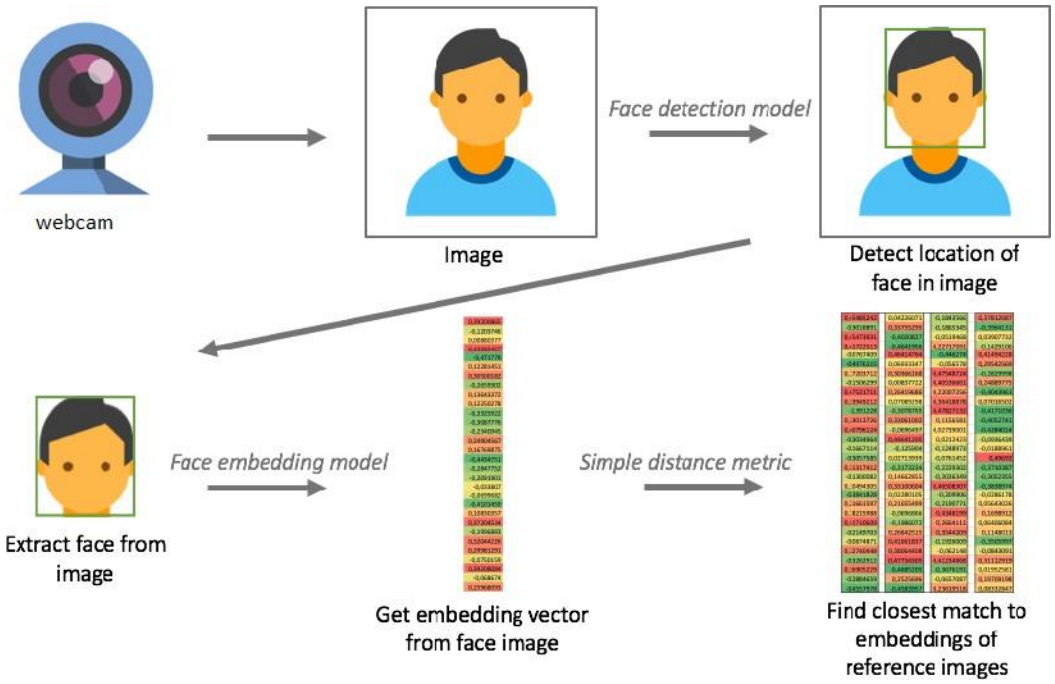


Figure 2.1: DETECTION METHOD

2.1) FEATURE BASE APPROACH:

This section gives an overview on the major human face recognition techniques that apply mostly to frontalfaces. The methods considered areHaar-cascade frontalface, Eigenfaces, Fisher-face, Local Binary Patterns Histograms.

A. Face Detection using Haar Cascade:

A hair cascade is defined as: a sequence of "square shaped" functions which together form a family of wavelets or a base. It is also focused on "Haar Wavelets," which organize pixels on the picture into squares, based on a hair wavelet approach

which was suggested in the 2001 "rapid object detection using an enhanced cascade of simple features" paper by Paul Viola and Michael Jones (P.Viola, 20010[11]. It is a learning approach based on computers, where many positive and negative representations are used to construct a cascade function. This is then used to detect objects Using "integral image" principles in order to compute "features" identified by the hair cascades.

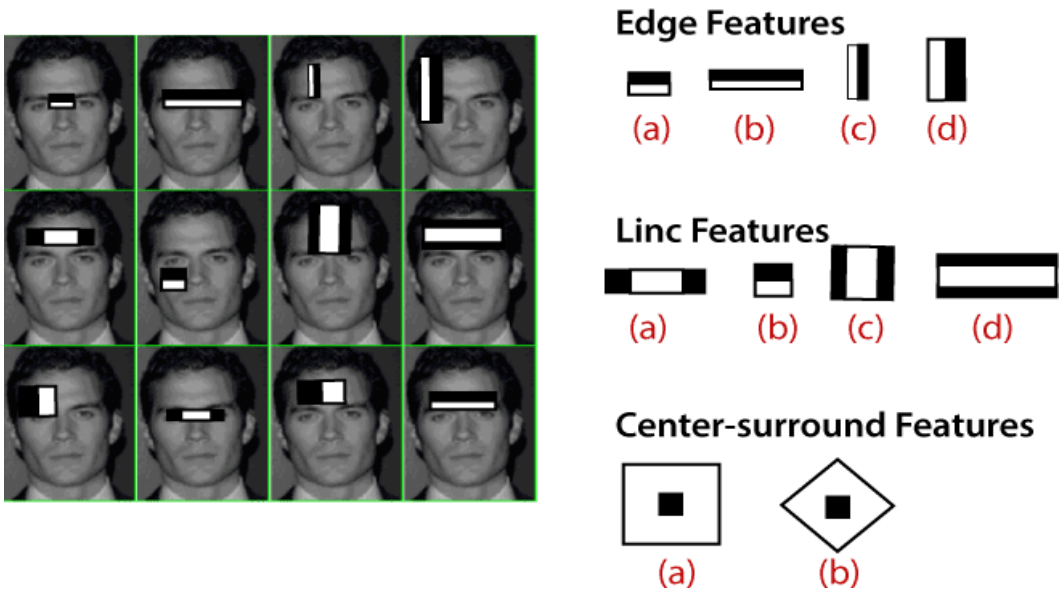


Figure: Extraction features in Harr-like features

B. FACE RECOGNITION USING EIGEN FACES:

Eigenfaces are images that can be added to a mean (average) face to create new facial images. We can write this mathematically as,

$$F = F_m + \sum_{i=1}^{n} \alpha_i F_i$$

where,

F is a new face.
F_m is the mean or the average face,
F_i is an EigenFace,
α_i are scalar multipliers we can choose to create new faces. They can be positive or negative.

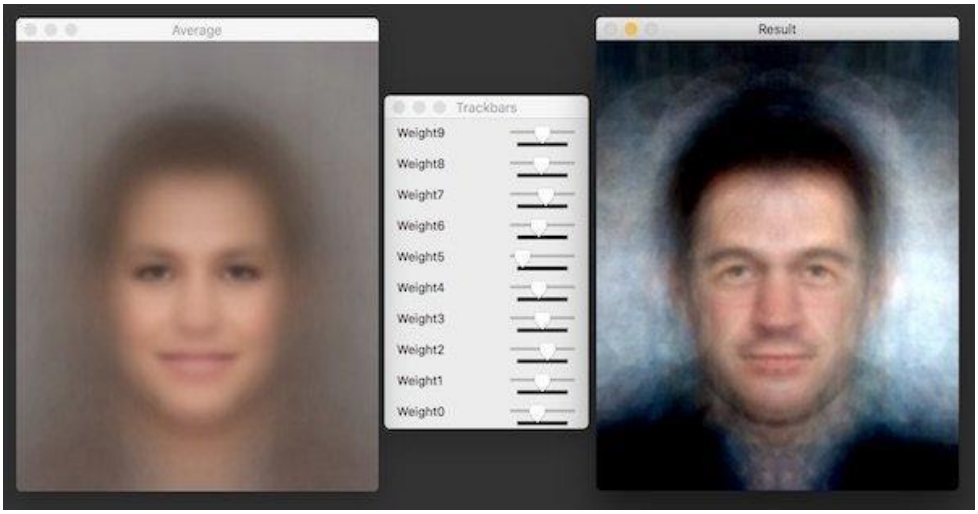


Figure : On the left is the mean image. On the right is a new face produced by adding 10 Eigenfaces with different weights (shown in center).

C. Face Recognition Using Fisherfase:

The most commonly known DAs, which can originate from an RA concept, is the Linear Discriminant Analysis (LDA). In 1936, Fisher. If LDAs are used to locate a subspace image of a set of face images, they use the Eigenface method to describe the space is known as Fisher's Linear Discriminant Analyzes (FLDA or LDA).

D. Face Recognition Using LBP:

Local Binary Pattern (LBP) Is an easy, but very strong user of texture who marks the image's pixels with each pixel's proximity, and views the outcome as binary numbers.It was first defined in 1994 (LBP) and is a powerful component for the classification of texture since then. In addition, LBP has been calculated to increase significantly the detection efficiency on certain datasets when coupled with histograms of oriented gradients (HOG) descriptor. By using LBP combined with histogram, face pictures with a simple data vector can be described.

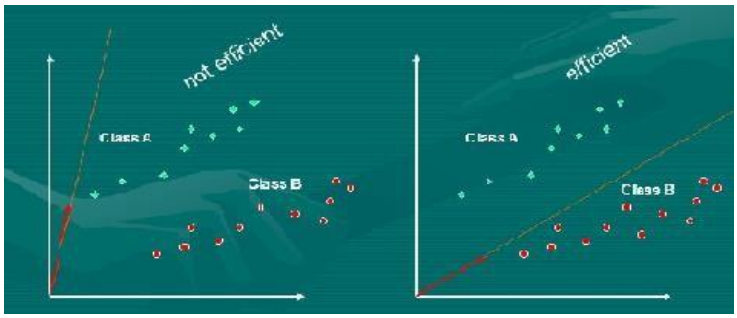


Fig: -PCA seeks directions that are efficient for representing the data.

Local Binary Patterns, or LBPs for short, the work popular of Ojala et al as texture descriptor in its 2002 paper, Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns (the definition of LBPs was implemented at the beginning of 1993). It computes a global representation of texture based on the Gray Level Co-occurrence Matrix, unlike Haralick. LBPs compute a regional texture representation. The binary image is built by comparing every pixel with the pixel surrounding area. For example, look at the original 3 x 3-pixel neighborhood LBP descriptor operated just same this:

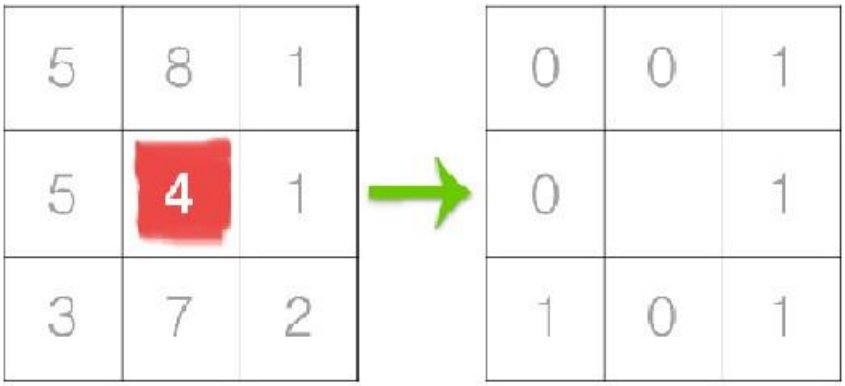
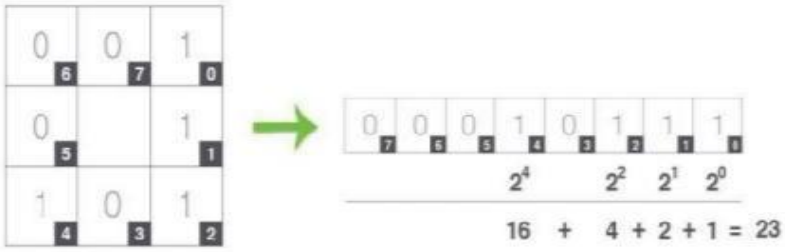


Fig6- Binary Pattern

A middle pixel and an 8-binary numbers cap is the first step in creating an LBP for the 8 Pixels neighborhood. In the figure above, we choose the center pixel and set it against its 8pixel neighborhood. If the pixel center is more than the following line, set the value to l; set the value to 0, otherwise. A total combination of 2 ^8= 256 possible LBP codes with eight pixels will be achieved. Thanks to a neighborhood of 3x3, we have 8 neighbors on which we must check binarily. This binary search results were stored in an 8-bit array and converted into decimal.



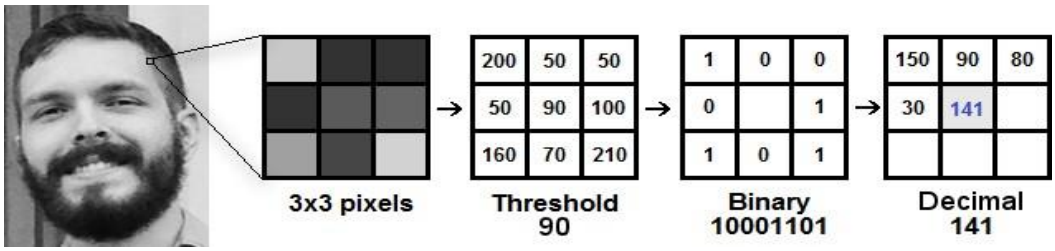


Fig: Face Recognition of LBPH Algorithm

2.2) LOW LEVEL ANALYSIS:

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

Fig Face Detection

2.3) 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. Besides face regions, facial features can be located by frame differences.

2.3.1 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 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 behavior 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.

2.3.2 Edge Base:

Face detection based on edges was introduced by Sakai et al. This work was based on analyzing 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. 'swork 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 image is constructed from the enhanced image by applying Sobel 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.

2.4 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.

2.4.1 Feature Searching:

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 [39] 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. 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.

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

The plural means:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

Real part:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

Imaginary part:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

among them:

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

Correspondence between the parameters in the code and the Gabor function parameters

The code implements the real part of the Gabor filter. The parameters in the code correspond to the real part parameters of the Gabor function as follows:

```
np.exp(-(_x**2 + Gamma**2 * _y**2) / (2 * Sigma**2)) * np.cos(2*np.pi*_x/Lambda + Psi)
```

correspond

$$\exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

2.5 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. 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. Image Base Approach.

2.5.1 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. 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. 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. The idea is based on [48] 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 frontalview faces and another one is used to detect faces turned up to 60 degrees to the left and right of the frontal view

2.6 LINEAR SUB SPACE METHOD:

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. A reasonable likeness (i.e.,capturing 95 percent of Eigen faces Method:An early example of employing eigen

vectors 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. The idea is arguably proposed first by Pearson in 1901 and then byHOTELLING in 1933 .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 eigen vectors of the covariance the variance).

2.7 STATISTICAL APPROCH:

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 Osunaet al. developed an efficient method to train an SVM for 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 . SVMs have also been used to detect faces and pedestrians in the wavelet domain.

CHAPTER-3

FACE DETECTION

`The problem of face recognition is all about face detection. This is a fact that seems quite bizarre to new researchers in this area. However, before face recognition is possible, one must be able to reliably find a face and its landmarks. This is essentially a segmentation problem and in practical systems, most of the effort goes into solving this task. In fact the actual recognition based on features extracted from these facial landmarks is only a minor last step. There are two types of face detection problems:

1) Face detection in images and

2) Real-time face detection

3.1 FACE DETECTION IN IMAGES

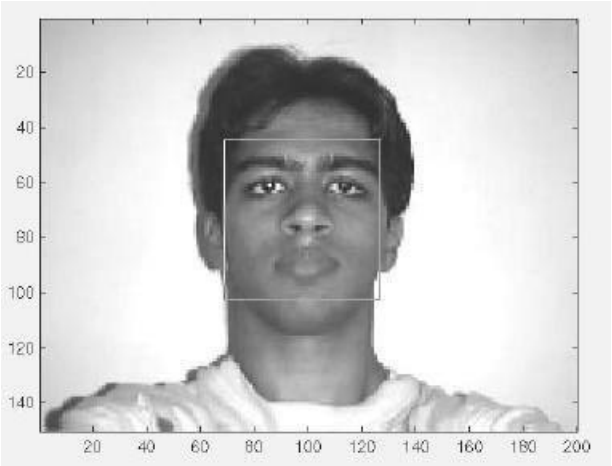


Figure 5.1 A successful face detection in an image with a frontal view of a huma face.

Most face detection systems attempt to extract a fraction of the whole face, thereby eliminating most of the background and other areas of an individual's head such as hair that are not necessary for the face recognition task. With static images, this is often done by running a across the image. The face detection system then judges if a face is present inside the window (Brunelli and Poggio, 1993). Unfortunately, with static images there is a very large search space of possible locations of a face in an image. Most face detection systems use an example based learning approach to decide

whether or not a face is present in the window at that given instant (Sung and Poggio,1994 and Sung,1995). A neural network or some other classifier is trained using supervised learning with 'face' and 'nonface' examples, thereby enabling it to classify an image (window in face detection system) as a 'face' or 'non-face'. Unfortunately, while it is relatively easy to find face examples, how would one find a representative sample of images which represent non-faces (Rowley et al., 1996)? Therefore, face detection systems using example-based learning need thousands of 'face' and 'non face' images for effective training. Rowley, Baluja, and Kanade (Rowley et al.,1996) used 1025 face images and 8000 non-face images (generated from 146,212,178 sub-images) for their training set! There is another technique for determining whether there is a face inside the face detection system's window - using Template Matching. The difference between a fixed target pattern (face) and the window is computed and thresholder. If the window contains a pattern which is close to the target pattern(face) then the window is judged as containing a face. An implementation of template matching called Correlation Templates uses a whole bank of fixed sized templates to detect facial features in an image (Bichsel, 1991 &Brunelli and Poggio, 1993). By using several templates of different (fixed) sizes, faces of different scales (sizes) are detected. The other implementation of template matching is using a deformable template (Yuille, 1992). Instead of using several fixed size templates, we use a deformable template (which is non-rigid) and there by change the size of the template hoping to detect a face in an image. A face detection scheme that is related to template matching is image invariants. Here the fact that the local ordinal structure of brightness distribution of a face remains largely unchanged under different illumination conditions (Sinha, 1994) is used to construct a spatial template of the face which closely corresponds to facial features. In other words, the average grey-scale intensities in human faces are used as a basis for face detection. For example, almost always an individual’s eye region is darker than his forehead or nose. Therefore, an image will match the template if it satisfies the 'darker than' and 'brighter than' relationships (Sung and Poggio, 1994).

3.2 REAL-TIME FACE DETECTION

Real-time face detection involves detection of a face from a series of frames from a videocapturing device. While the hardware requirements for such a system are far more stringent, from a computer vision stand point, real-time face detection is actually a far simpler process than detecting a face in a static image. This is because unlike most of our surrounding environment, people are continually moving. We walk around, blink, fidget, wave our hands about, etc.

Figure 3.2.1: Frame 1 from camera

Figure 3.2.2: Frame 2 from camera

Figure 3.2.3: Spatio-Temporally filtered image

Since in real-time face detection, the system is presented with a series of frames in which to detect a face, by using spatiotemporal filtering (finding the difference between subsequent frames), the area of the frame that has changed can be identified and the individual detected (Wang and Adelson, 1994 and Adelson and Bergen 1986).Further more as seen in Figure exact face locations can be easily identified by using a few simple rules, such as,

1) the head is the small blob above a larger blob -the body

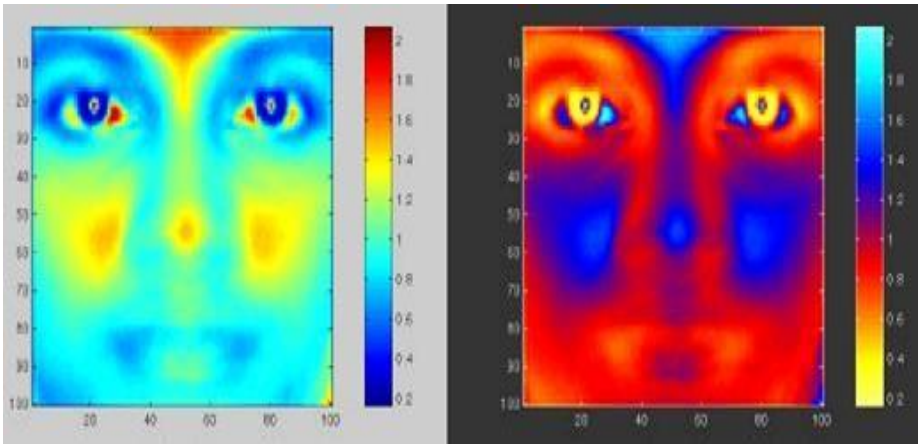
2) head motion must be reasonably slow and contiguous -heads won't jump around erratically (Turk and Pentland 1991a, 1991b).

Real-time face detection has therefore become a relatively simple problem and is possible even in unstructured and uncontrolled environments using these very simple image processing techniques and reasoning rules.

3.3 FACE DETECTION PROCESS

Fig 3.3: Face Detection

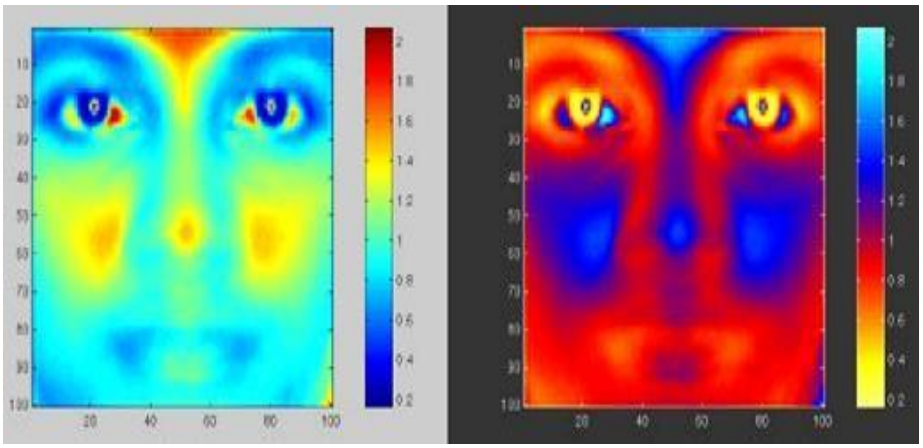
It is process of identifying different parts of human faces like eyes, nose, mouth, etc this process can be achieved by using python code. In this project the author will attempt to detect faces in still images by using image invariants. To do this it would be useful to study the grey scale intensity distribution of an average human face. The following 'average human face' was constructed from a sample of 30 frontal view human faces, of which 12 were from females and 18 from males. A suitably scaled colormap has been used to highlight grey-scale intensity differences.



Scaled Colourmap Scaled Colourmap(negative)

Figure 3.3.1 Average human face in grey-scale

The grey-scale differences, which are invariant across all the sample faces are strikingly apparent. The eye-eyebrow area seem to always contain dark intensity (low) gray-levels while nose forehead and cheeks contain bright intensity (high) grey-levels. After a great deal of experimentation, the researcher found that the following areas of the human face were suitable for a face detection system based on image invariants and a deformable template.



Scaled Colourmap Scaled Colourmap (negative)

Figure 3.3.2 Area chosen for face detection (indicated on average human face in gray scale)

The above facial area performs well as a basis for a face template, probably because of the clear divisions of the bright intensity invariant area by the dark intensity invariant regions. Once this pixel area is located by the face detection system, any particular area required can be segmented based on the proportions of the average human face After studying the above images it was subjectively decided by the author to use the following as a basis for dark intensity sensitive and bright intensity sensitive templates. Once these are located in a subject's face, a pixel area 33.3% (of the width of the square window) below this.

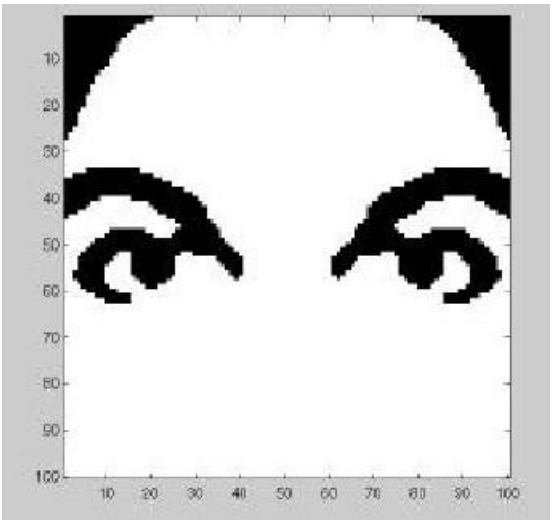


Figure 3.3.3: Basis for a bright intensity invariant sensitive template.

Note the slight differences which were made to the bright intensity invariant sensitive template which were needed because of the pre-processing done by the system to overcome irregular lighting (chapter six). Now that a suitable dark and bright intensity invariant templates have been decided on, it is necessary to find a way of using

these to make 2 A-units for a perceptron, i.e. a computational model is needed to assign neurons to the distributions displayed.

San window over image

Classify window as either

1. Face

2. Non – Face

3.4FACE DETECTION ALGORITHM

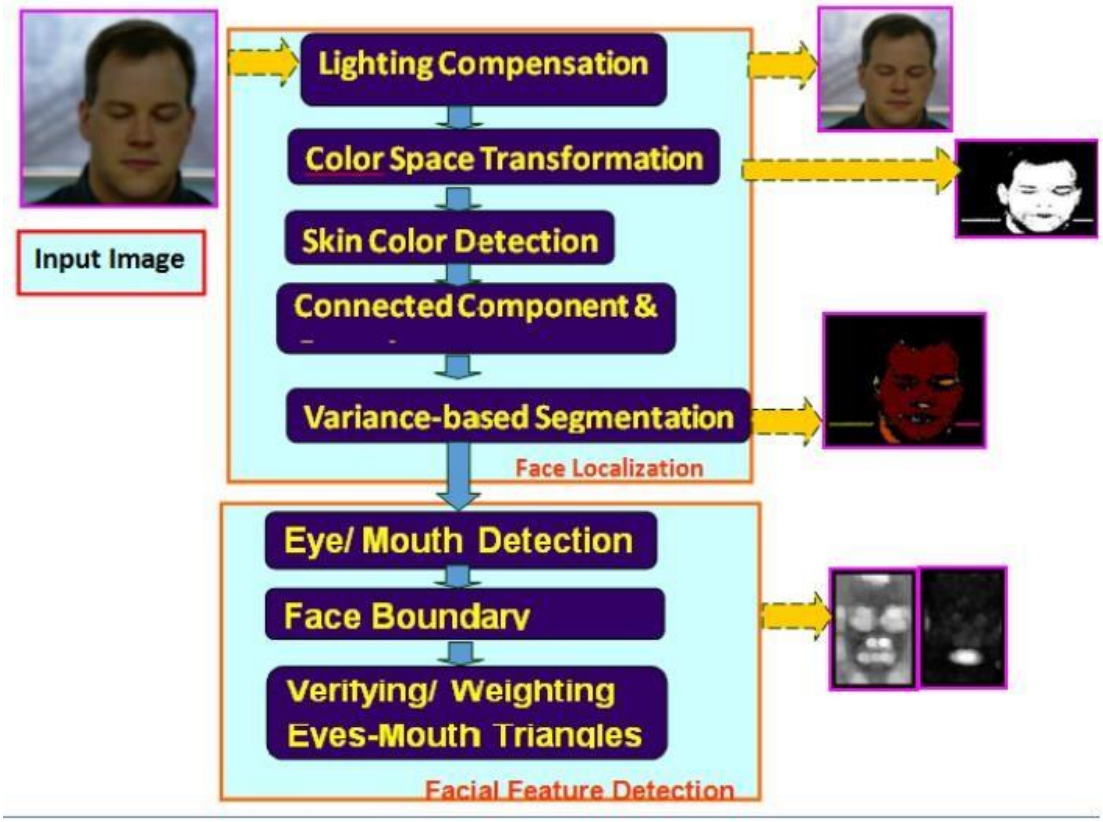


Fig 3.4 Face detection algorithm

Face Detection with Open CV-Python:

Now we have a fair idea about the intuition and the process behind Face recognition. Let us now use Open CV library to detect faces in an image.

Load the necessary Libraries

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
```

Loading the image to be tested in grayscale

We shall be using the image below:

DISPLAYING THE IMAGE:

```
#Loading the image to be tested
test_image = cv2.imread('data/baby1.jpg')

#Converting to grayscale
test_image_gray = cv2.cvtColor(test_image,
cv2.COLOR_BGR2GRAY)

# Displaying the grayscale image
plt.imshow(test_image_gray, cmap='gray')
```

Since we know that OpenCV loads an image in BGR format, so we need to convert it into RBG format to be able to display its true colors. Let us write a small function for that.

```
<matplotlib.image.AxesImage at 0x11bb375f8>
```

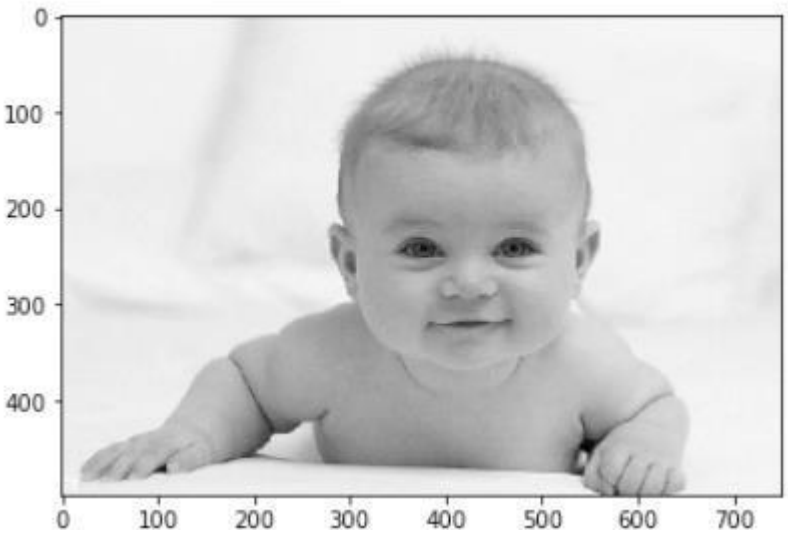


IMAGE OBTAINED

CHAPTER-4

FACE RECOGNITION

Over the last few decades many techniques have been proposed for face recognition. Many of the techniques proposed during the early stages of computer vision cannot be considered successful, but almost all of the recent approaches to the face recognition problem have been creditable. According to the research by Brunelli and Poggio (1993) all approaches to human face recognition can be divided into two strategies:

(1) Geometrical features and

(2) Template matching.

4.1 FACE RECOGNITION USING GEOMETRICAL FEATURES:

This technique involves computation of a set of geometrical features such as nose width and length, mouth position and chin shape, etc. from the picture of the face we want to recognize. This set of features is then matched with the features of known individuals. A suitable metric such as Euclidean distance (finding the closest vector) can be used to find the closest match. Most pioneering work in face recognition was done using geometric features (Kanade, 1973), although Craw et al. (1987) did relatively recent work in this area.

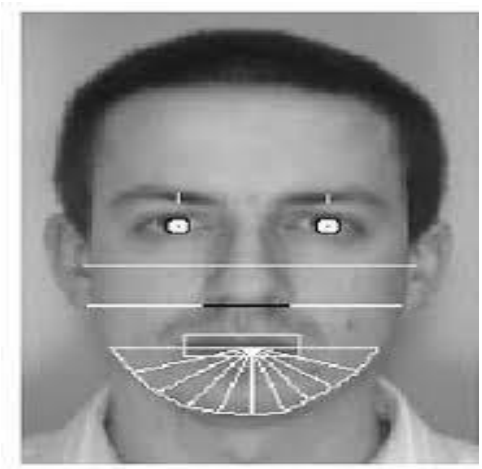


Figure 6.1 Geometrical features (white) which could be used for face recognition

The advantage of using geometrical features as a basis for face recognition is that recognition is possible even at very low resolutions and with noisy images (images with many disorderly pixel intensities). Although the face cannot be viewed in detail its overall geometrical configuration can be extracted for face recognition. The technique's main disadvantage is that automated extraction of the facial geometrical features is very hard. Automated geometrical feature extraction-based recognition is also very sensitive to the scaling and rotation of a face in the image plane (Brunelli and Poggio, 1993). This is apparent when we examine Kanade's(1973) results where he reported a recognition rate of between 45-75 % with a database of only 20 people. However, if

these features are extracted manually as in Goldstein et al. (1971), and Kaya and Kobayashi (1972) satisfactory results may be obtained.

4.1.1 Face recognition using template matching:

This is similar the template matching technique used in face detection, except here we are not trying to classify an image as a 'face' or 'non-face' but are trying to recognize a face.

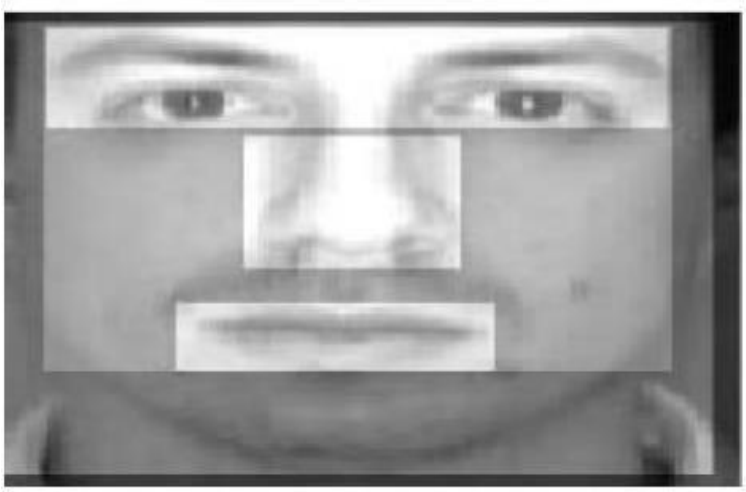


Figure : 4.1.1 Face recognition using template matching

Whole face, eyes, nose and mouth regions which could be used in a template matching strategy. The basis of the template matching strategy is to extract whole facial regions (matrix of pixels) and compare these with the stored images of known individuals. Once again Euclidean distance can be used to find the closest match. The simple technique of comparing grey-scale intensity values for face recognition was used by Baron (1981). However, there are far more sophisticated methods of template matching for face recognition. These involve extensive preprocessing and transformation of the extracted grey-level intensity values. For example, Turk and Pentland (1991a) used Principal Component Analysis, sometimes known as the eigenfaces approach, to pre-process the gray-levels and Wiskott et al. (1997) used Elastic Graphs encoded using Gabor filters to pre-process the extracted regions. An investigation of geometrical features versus template matching for face recognition by Brunelli and Poggio (1993) came to the conclusion that although a feature-based strategy may offer higher recognition speed and smaller memory requirements, template-based techniques offer superior recognition accuracy.

Face is the crucial part of the human body that uniquely identifies a person. Using the face characteristics as biometric, the face recognition system can be implemented. The most demanding task in any organization is attendance marking. In traditional attendance system, the students are called out by the teachers and their presence or absence is marked accordingly. However, these traditional techniques are time consuming and tedious. In this project, the Open CV based face recognition

approach has been proposed. This model integrates a camera that captures an input image, an algorithm for detecting face from an input image, encoding and identifying the face, marking the attendance in a spreadsheet and converting it into PDF file. The training database is created by training the system with the faces of the authorized students. The cropped images are then stored as a database with respective labels. The features are extracted using LBPH algorithm.

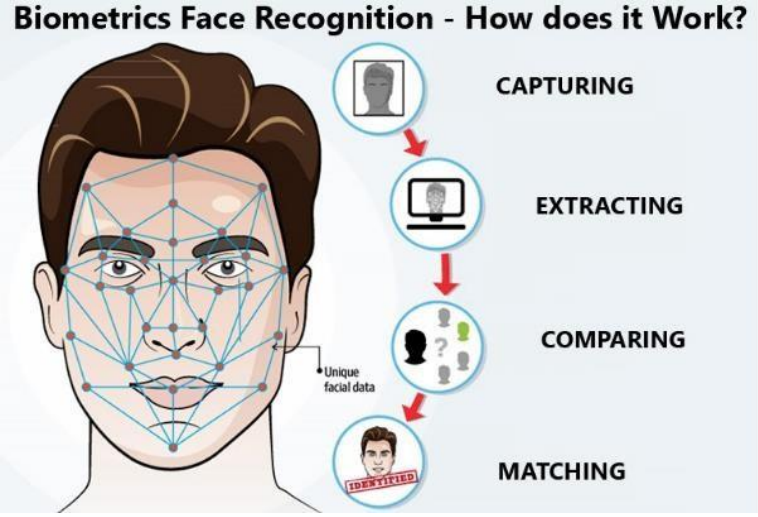


Fig: Face Recognition Technique

Face recognition systems use computer algorithms to pick out specific, distinctive details about a person's face. These details, such as distance between the eyes or shape of the chin, are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. The data about a particular face is often called a face template and is distinct from a photograph because it's designed to only include certain details that can be used to distinguish one face from another.

4.2IMAGE PROCESSING:

The facial recognition process can be split into two major stages: processing which occurs before detection involving face detection and alignment and later recognition is done using feature extraction and matching steps.

1. FACE DETECTION:

The primary function of this step is to conclude whether the human faces emerge in a given image, and what is the location of these faces. The expected outputs of this step are patches which contain each face in the input image. In order to get a more

robust and easily design-able face recognition system.Face alignment is performed to rationalize the scales and orientation of these patches.

2.FEATURE EXTRACTION:

Following the face detection step the extraction of human face patches from images is done. After this step, the conversion of face patch is done into vector with fixed coordinates or a set of landmark points.

3.FACE RECOGNITION:

The last step after the representation of faces is to identify them. For automatic recognition we need to build a face database. Various images are taken foe each person and their features are extracted and stored in the database. Then when an input image is fed the face detection and feature extraction is performed and its feature to each face class is compared and stored in the database.

4. ALGORITHM:

There are various algorithms used for facial recognition. Some of them are as follows:

1.Eigen faces

2.Fisher faces

3.Local binary patterns histograms

1.EIGEN FACES:

This method is a statistical plan. The characteristic which influences the images is derived by this algorithm. The whole recognition method will depend on the training database that will be provided. The images from two different classes are not treated individually.

2.FISHER FACES:

Fisher faces algorithm also follows a progressive approach just like the Eigen faces. This method is a alteration of Eigen faces so it uses the same principal Components Analysis. The major conversion is that the fisher faces considers the classes. As mentioned previously, the Eigen faces does not differentiate between the two pictures from two differed classes while training. The total average affects each picture. A Fisher face employs Linear Discriminant Analysis for distinguishing between pictures from a different class.

3.LOCAL BINARY PATTERNS HISTOGRAMS:

This method needs the gray scale pictures for dealing with the training part. This algorithm in comparison to other algorithms is not a holistic approach.

PARAMETERS:

LBPH uses the following parameters:

1.Radius:

Generally, 1 is set as a radius for the circular local binary pattern which denotes the radius around the central pixel.

2. Neighbour:

The number of sample points surrounding the central pixel which is generally 8. The computational cost will increase with increase in number of sample points.

3. Grid X:

The number of cells along the horizontal direction is represented as Grid X. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

4. Grid Y:

The number of cells along the vertical direction is represented as Grid Y. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

The number of cells along the vertical direction is represented as Grid Y. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

4.3 Face Recognition Based Attendance Marking System:

The second research journals "Face Recognition Based Attendance Marking System" (SenthamilSelvi, Chitrakala, Antony Jenitha, 2014) is based on the identification of face recognition to solve the previous attendance system‟s issues. This system uses camera to capture the images of the employee to do face detection and recognition. The captured image is compared one by one with the face database to search for the worker‟s face where attendance will be marked when a result is found in the face database. The main advantage of this system is where attendance is marked on the server which is highly secure where no one can mark the attendance of other. Moreover, in this proposed system, the face detection algorithm is improved by using the skin classification technique to increase the accuracy of the detection process. Although more efforts are invested in the accuracy of the face detection algorithm, the system is yet not portable. This system requires a standalone computer which will need a constant power supply that makes it not portable. This type of system is only suitable for marking staff‟s attendance as they only need to report their presence once a day, unlike students which require to report their attendance at every class on a particular

day, it will be inconvenient if the attendance marking system is not portable. Thus, to solve this issue, the whole attendance management system can be developed on an portable module so that it can be work just by executing the python program.

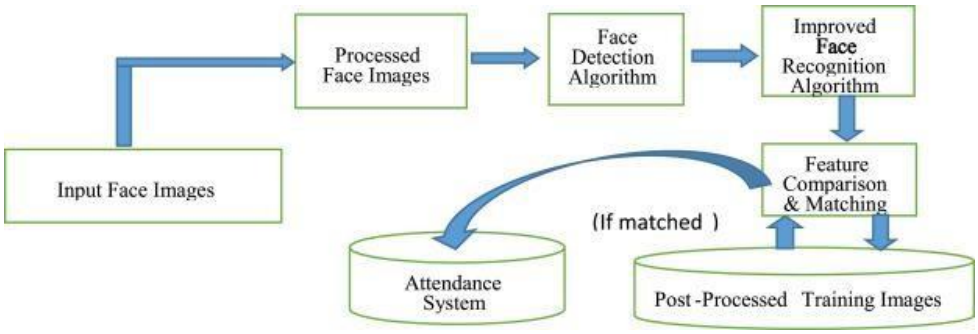


Fig:Face Recognition Based Attendance Marking System

4.4 Face Recognition Based Detailed Report:

At the present period of time, security is as a matter-of-course for anyone, whether it's data security or security of their organization. With the technological advancement and the use of IOT is increasing day by day. In some large organizations there are many number of people working if we want a particular person bio data, we can access the data through face recognition instead of checking manually. Organization contains huge no of students or employees, it is difficult to distinguish the stranger & associate(organized person) so we can overcome this problem by using face detection Bio-data.

Using Web Cam Face Detection we can record multifarious faces and we can get detailed report of a particular person using their Face Id's. It requires no physical interaction on behalf of the user. It is accurate and allows for high enrolment and verification rates. It can use your existing hardware infrastructure, existing cameras and image capture devices will work with no problems.

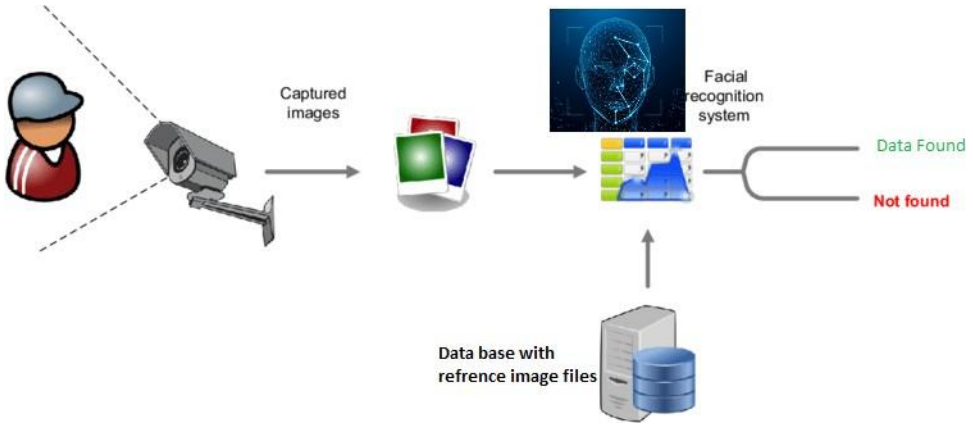


Figure: Face Recognition Based Detailed Report

4.5 PROBLEM SCOPE AND SYSTEM SPECIFICATION:

The following problem scope for this project was arrived at after reviewing the literature on face detection and face recognition, and determining possible real-world situations where such systems would be of use. The following system(s) requirements were identified.

1. A system to detect frontal view faces in static images

2. A system to recognize a given frontal view face

3. Only expressionless, frontal view faces will be presented to the face detection &recognition

4. All implemented systems must display a high degree of lighting invariency.

5. All systems must posses near real-time performance.

6. Both fully automated and manual face detection must be supported

7. Frontal view face recognition will be realised using only a single known image

8. Automated face detection and recognition systems should be combined into a fully automated face detection and recognition system. The face recognition sub-system must display a slight degree of invariency to scaling and rotation errors in the segmented image extracted by the face detection sub-system.

9. The frontal view face recognition system should be extended to a pose invariant face recognition system. Unfortunately although we may specify constricting conditions to our problem domain, it may not be possible to strictly adhere to these conditions when implementing a system in the realworld.

4.6 BRIEF OUT LINE OF THE IMPLEMENTED SYSTEM

Fully automated face detection of frontal view faces is implemented using a deformable template algorithm relying on the image invariants of human faces. This was chosen because a similar neural-network based face detection model would have needed far too much training data to be implemented and would have used a great deal of computing time. The main difficulties in implementing a deformable template-based technique were the creation of the bright and dark intensity sensitive templates and designing an efficient implementation of the detection algorithm.

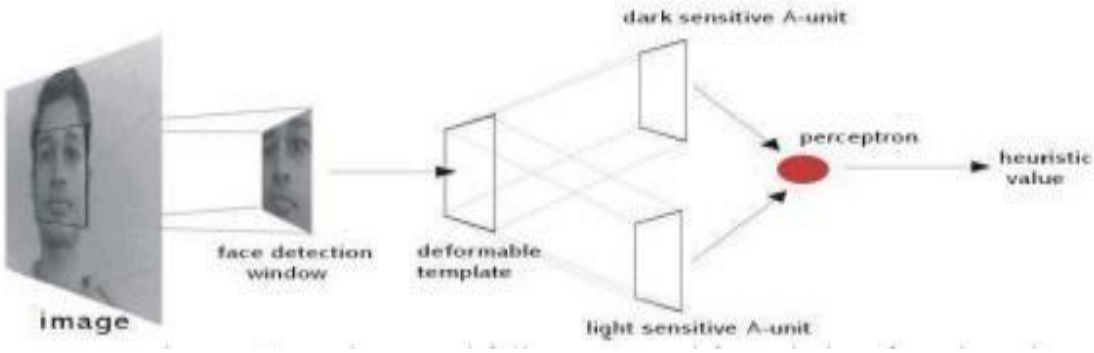


Figure 4.6 Implemented fully automated frontal view face detection model

A manual face detection system was realized by measuring the facial proportions of the average face, calculated from 30 test subjects. To detect a face, a human operator would identify the locations of the subject's eyes in an image and using the proportions of the average face, the system would segmentation an area from the image.

A template matching based technique was implemented for face recognition. This was because of its increased recognition accuracy when compared to geometrical features based techniques and the fact that an automated geometrical features based technique would have required complex feature detection pre-processing.

Of the many possible template matching techniques, Principal Component Analysis was chosen because it has proved to be a highly robust in pattern recognition tasks and because it is relatively simple to implement. The author would also liked to have implemented a technique based on Elastic Graphs but could not find sufficient literature about the model to implement such a system during the limited time available for this project.

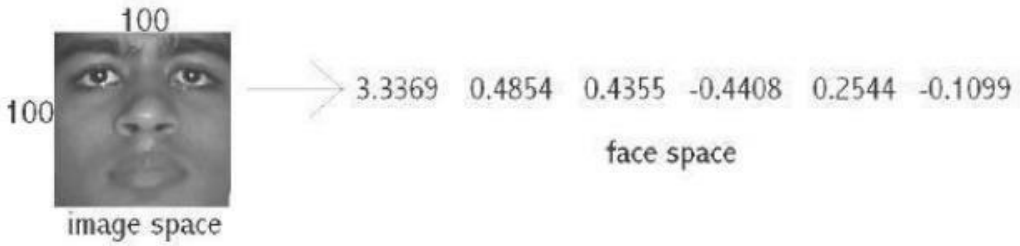


Figure 4.5.1: Principal Component Analysis transform from 'image space' to 'face space'.

Using Principal Component Analysis, the segmented frontal view face image is transformed from what is sometimes called 'image space' to 'face space'. All faces in the face database are transformed into face space. Then face recognition is achieved by

transforming any given test image into face space and comparing it with the training set vectors. The closest matching training set vector should belong to the same individual as the test image. Principal Component Analysis is of special interest because the transformation to face space is based on the variation of human faces (in the training set). The values of the 'face space' vector correspond to the amount certain 'variations' are present in the test image.

Face recognition and detection system is a pattern recognition approach for personal identification purposes in addition to other biometric approaches such as fingerprint recognition, signature, retina and so forth. Face is the most common biometric used by humans applications ranges from static, mug-shot verification in a cluttered background.

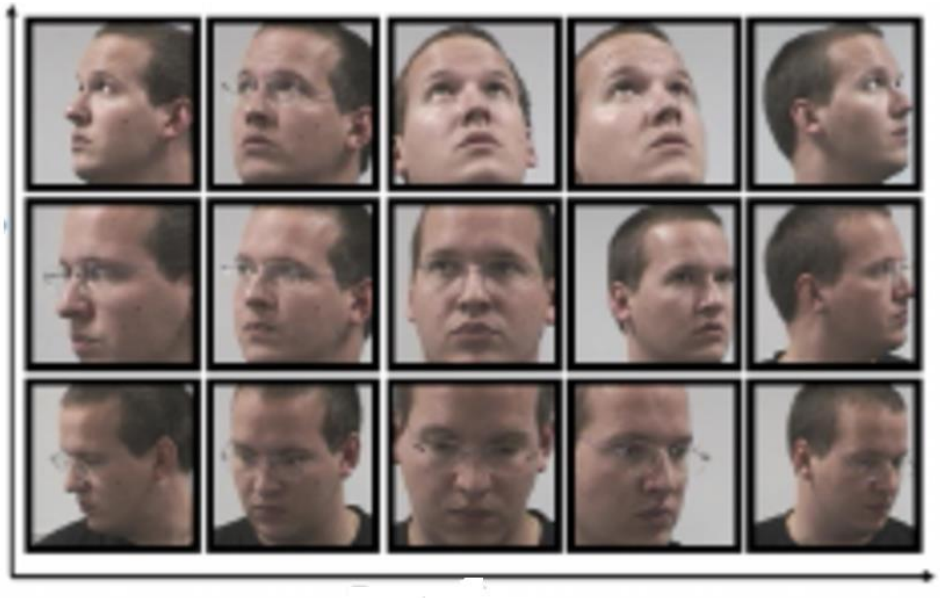


Fig 4.5.2 Face Recognition

4.7 FACE RECOGNITION DIFFICULTIES:

1. Identify similar faces (inter-class similarity)

2. Accommodate intra-class variability due to

2.1 head pose

2.2 illumination conditions

2.3 expressions

2.4 facial accessories

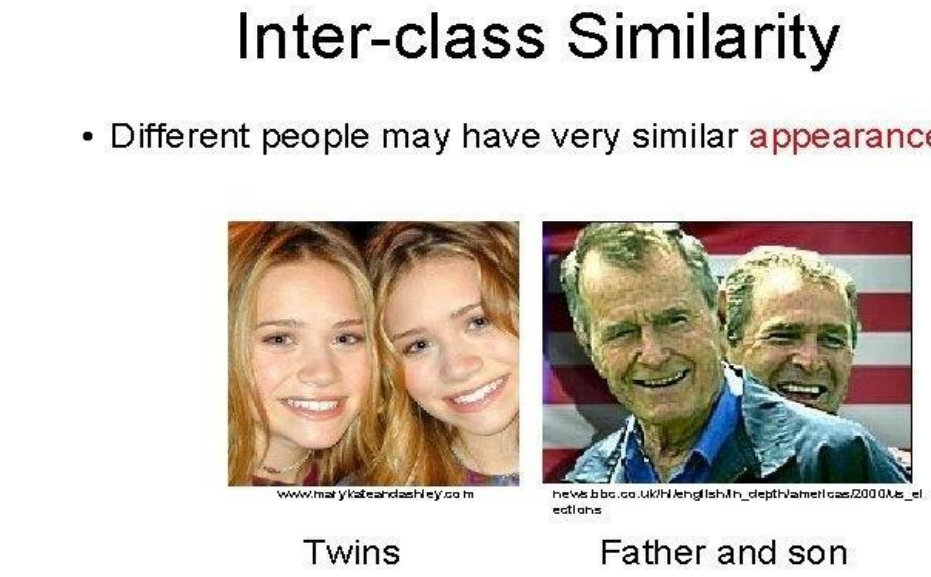
2.5 aging effects

3. Cartoon faces

Inter - class similarity:

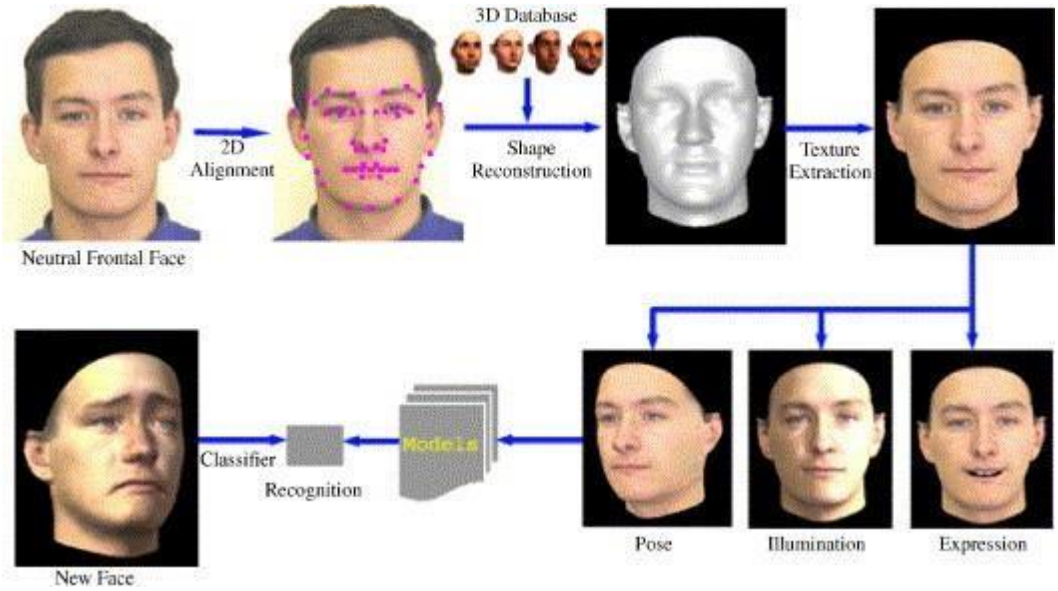
- Different persons may have very similar appearance

Face recognition and detection system is a pattern recognition approach for personal identification purposes in addition to other biometric approaches such as fingerprint recognition, signature, retina and so forth. 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 images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.



4.8 IMPROVING FACE DETECTION USING RECONSTRUCTION:

Reconstruction cannot be used as a means of face detection in images in near real-time since it would involve resizing the face detection window area and large matrix multiplication, both of which are computationally expensive. However, reconstruction can be used to verify whether potential face locations identified by the deformable template algorithm actually contain a face. If the reconstructed image differs greatly from the face detection window then the window probably does not contain a face. Instead of just identifying a single potential face location, the face detection algorithm can be modified to output many high 'faceness' locations which can be verified using reconstruction. This is especially useful because occasionally the best 'faceness' location found by the deformable template algorithm may not contain the ideal frontal view face pixel area.



potential face locations that have been identified by the face detection system (the best face locations it found on its search) are checked whether they contain a face. If the threshold level (maximum difference between reconstruction and original for the original to be a face) is set correctly this will be an efficient way to detect a face. The deformable template algorithm is fast and can reduce the search space of potential face locations to a handful of positions. These are then checked using reconstruction. The number of locations found by the face detection system can be changed by getting it to output, not just the best face locations it has found so far but any location, which has a 'faceness' value, which for example is, at least 0.9 times the best heuristic value that has been found so far. Then there will be many more potential face locations to be checked using reconstruction. This and similar speed versus accuracy trade-off decisions have to be made keeping in mind the platform on which the system is implemented. Similarly, instead of using reconstruction to check the face detection system's output, the output's correlation with the average face can be checked. The segmented areas with a high correlation probably contains a face. Once again a threshold value will have to be established to classify faces from non-faces. Similar to reconstruction, resizing the segmented area and calculating its correlation with the average face is far too expensive to be used alone for face detection but is suitable for verifying the output of the face detection system.

4.9 POSE INVARIANT FACE RECOGNITION

Extending the frontal view face recognition system to a pose-invariant recognition system is quite simple if one of the proposed specifications of the face recognition system is relaxed. Successful pose-invariant recognition will be possible if many images of a known individual are in the face database. Nine images from each known individual can be taken as shown below. Then if an image of the same individual is submitted within a 30o angle from the frontal view he or she can be identified. Nine images in face database from a single known individual.

FIGURE: Unknown image from same individual to be identified

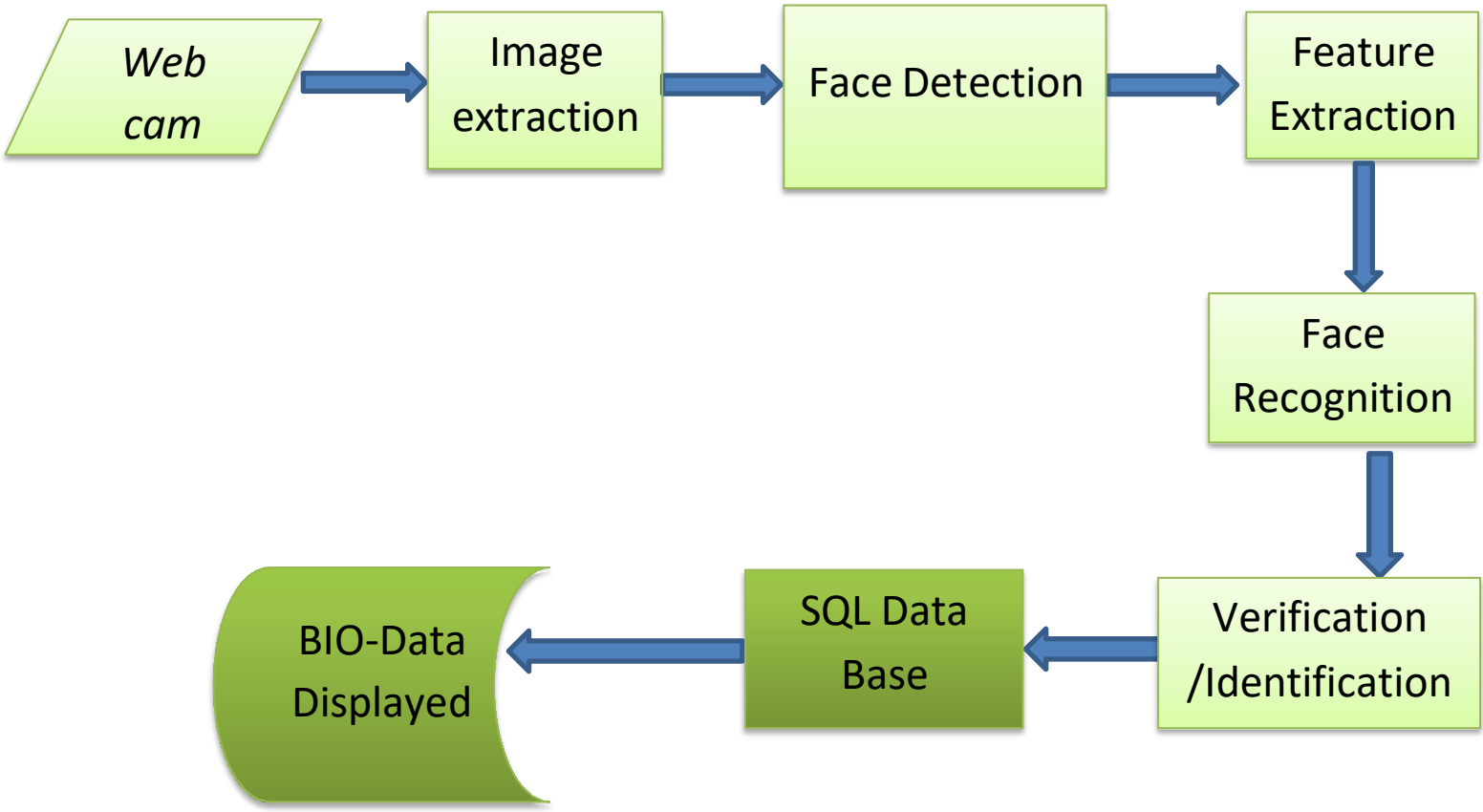
Fig: 4.9 Pose invariant face recognition.

Pose invariant face recognition highlights the generalization ability of PCA. For example, when an individual's frontal view and 30o left view known, even the individual's 15o left view can be recognized.

CHAPTER -5

Flow charts

5.1 Working overview:



WEB CAM:

Capture an image by accessing the webcam of your system and save it to your machine.

How to use:

- Run the file webcam-capture-v1.01.py by running the command python3 webcam-capture-v1.01.py
- The webcam will start running.
- Bring the picture that you want to save in the webcam frame.
- Once the object is in the right frame, press the key 's' to save a picture.
- If you want to quit, just press 'q'.

- After hitting 's' to save the picture, you will get a view of the saved image which will automatically close in 1.6s and a new image file will be created in the same directory as that of the program. The image will be saved as saved_image.jpg
- The file saved image is furthered converted to grayscale and then resized to 28x28 size for further use in relevant programs.

FACE DETECTION:

Face detection -- also called facial detection -- is an artificial intelligence (AI) based computer technology used to find and identify human faces in digital images. ... It now plays an important role as the first step in many key applications -- including face tracking, face analysis and facial recognition.

FEATURE EXTRACTION:

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables.

In real life, all the data we collect are in large amounts. To understand this data, we need a process. Manually, it is not possible to process them. Here's when the concept of feature extraction comes in.

Suppose you want to work with some of the big machine learning projects or the coolest and popular domains such as deep learning, where you can use images to make a project on object detection. Making projects on computer vision where you can to work with thousands of interesting project in the image data set. To work with them, you have to go for feature extraction procedure which will make your life easy.

FACE RECOGNITION:

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time. Law enforcement may also use mobile devices to identify people during police stops.

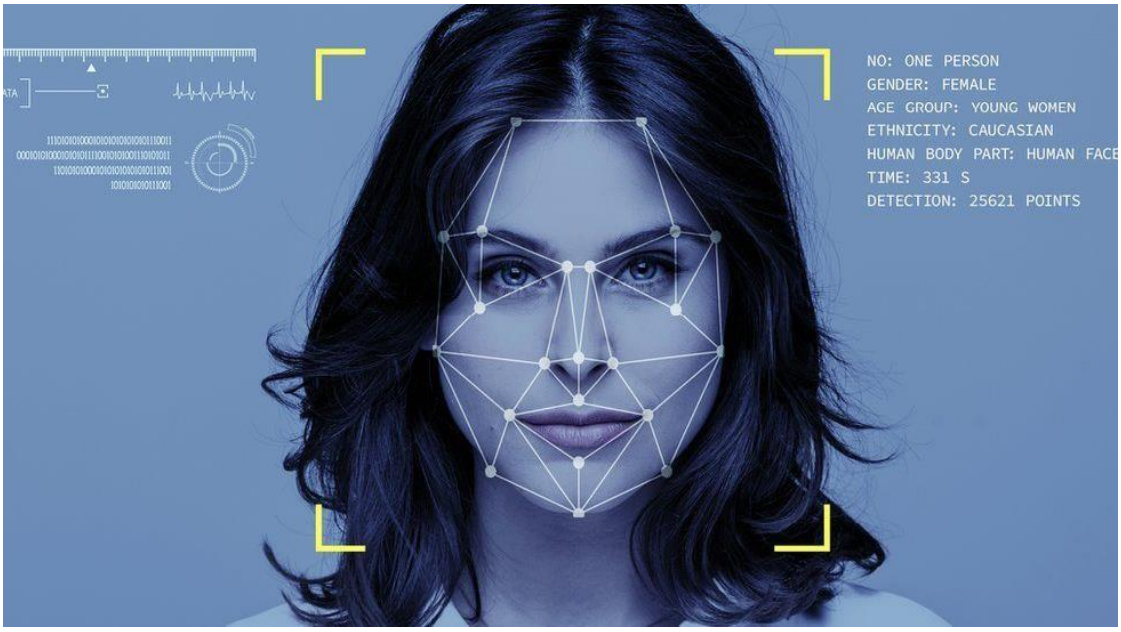


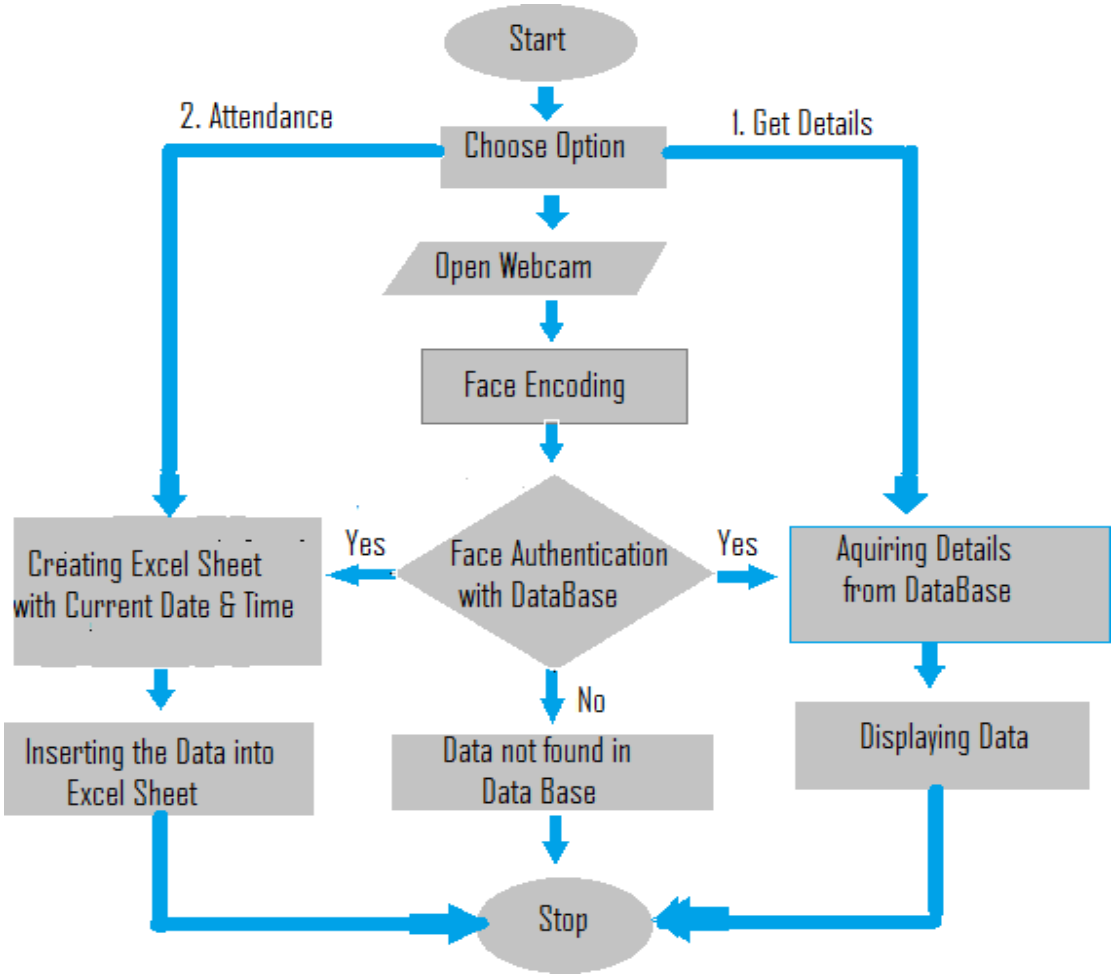
Fig. Capturing features

VERIFICATION/ IDENTIFICATION:

Facial verification technology confirms that your physical face matches the one in the ID document, and that you are real and completing this application right now. You know facial verification is happening. You choose to do it. There is a direct personal benefit to you (you get to go on your vacation). And with face verification technology, you know that the images are kept behind a privacy firewall and are subject to strict GDPR rules.

There is also face detection, which is the process of identifying when a human face is present in a video or image, without identifying that person in any way.

5.2 Program Flow chart:-



ALGORITHM:-

1. Code is programmed and it display the option to choose either details or attendance record .
2. Image is captured through webcam to recognize the person.
3. If face authentication is successful go to step4 otherwise goto step7.
4. Based on chosen option goto step5 or step6.
5. The details will acquire from database and it will be displayed.
6. Excel sheet will be created with current date i.e. attendance is recorded.
7. Data not found in database.

CHAPTER -6

. SOFTWARE DEVELOPMENT

Pre-requisites:

- Python(3.8 version) Environment
- Jupyter notebook and related libraries installations
- A Working Webcam
- SQL installations

6.1 Python(3.8 version) Environment:

While The Python Language Reference describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions. Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components. In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the Python Package Index.

The bulk of the library, however, consists of a collection of modules. There are many ways to dissect this collection. Some modules are written in C and built in to the Python interpreter; others are written in Python and imported in source form. Some modules provide interfaces that are highly specific to Python, like printing a stack trace; some provide interfaces that are specific to particular operating systems, such as access to specific hardware; others provide interfaces that are specific to a particular application domain, like the World Wide Web. Some modules are available in all versions and ports of Python; others are only available when the underlying system supports or requires them; yet others are available only when a particular configuration option was chosen at the time when Python was compiled and installed.

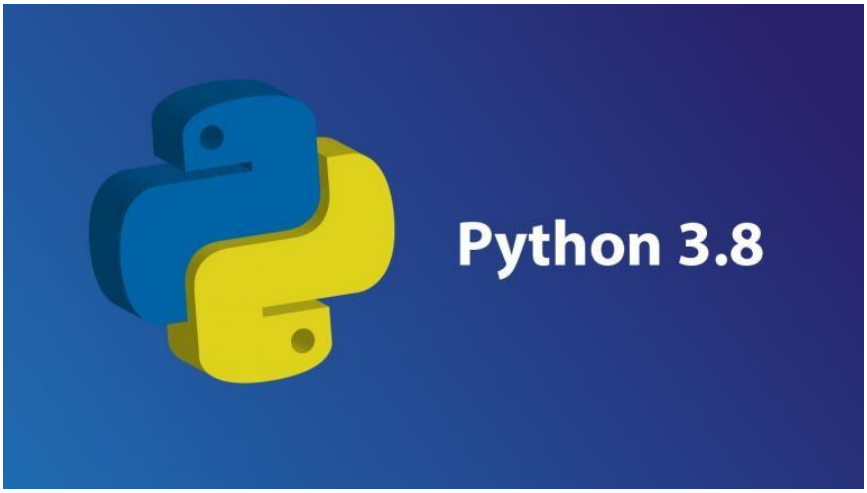


Figure 5.1: Python 3.8 Environment

6.2 Features of Jupyter Notebook:

1. **Ease of Use:** Facilitates simpler user experience through syntax highlighting, shortcuts, completion aid automation, etc.

2. **File Management:** Manage individual software component files as a part of the greater overall project.

3. **Multi-Language Support:** Provides the ability to write in more than one programming language.

4. **Customization:** Allows users to modify interface (e.g., theme selection) and layout by personal preference.

5. **Straight-Out-the-Box Functionality:** Performs core functionalities without the need of immediate additional feature or add-on installations.

6. **Help Guides:** Supplies users with basic-to-intermediate guides regarding IDE functionality and use cases.

7. **Patching & Update:** Downloads and installs updates automatically for product improvement and pace-keeping with programming language evolution.

6.3 Advantages Of Jupyter Notebook:

One of the most important advantages of Jupyter Notebooks is coming from the cell-by-cell nature of the notebooks, which means **splitting every possible logical step** allows exploring the data in hand in a very interactive manner:

- Every small cell can be deleted or modified without any impact on the rest of the analysis.
- Any change on the cell **can be reverted**, which allows quick movement without thinking too much about the consequences of the code.

Any input and output of a certain cell is **highly visible** from the previous steps and can be easily displayed before starting to explore the data.

6.4 Jupyter notebook and related libraries installations:

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at project jupyter.

Jupyter Notebooks are a spin-off project from the Python project, which used to have an Python Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Python, and Jupyter ships with the Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Installation:

If so, then you can use a handy tool that comes with Python called pip to install Jupyter Notebook like this:

```
Shell

$ pip install jupyter
```

The next most popular distribution of Python is Anaconda. Anaconda has its own installer tool called **conda** that you could use for installing a third-party package. However, Anaconda comes with many scientific libraries preinstalled, including the Jupyter Notebook, so you don't actually need to do anything other than install Anaconda itself.

Starting the Jupyter Notebook Server

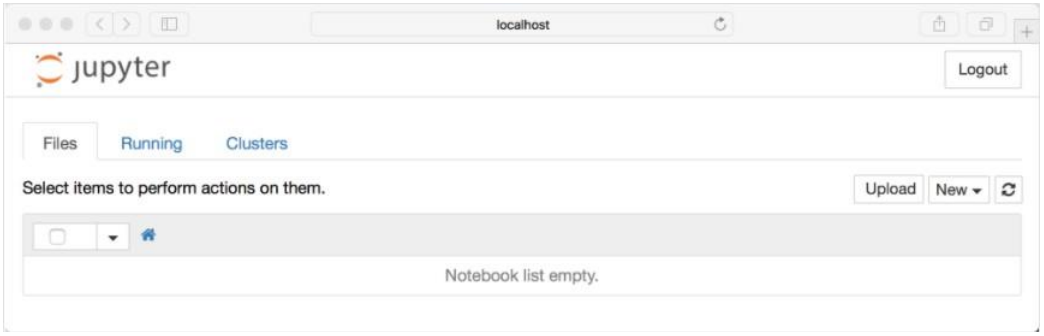
Now that you have Jupyter installed, let's learn how to use it. To get started, all you need to do is open up your terminal application and go to a folder of your choice. I recommend using something like your Documents folder to start out with and create a sub folder there called ***Notebooks*** or something else that is easy to remember.

Then just go to that location in your terminal and run the following command:

```
Shell

$ jupyter notebook
```

This will start up Jupyter and your default browser should start and browser should now look something like this:



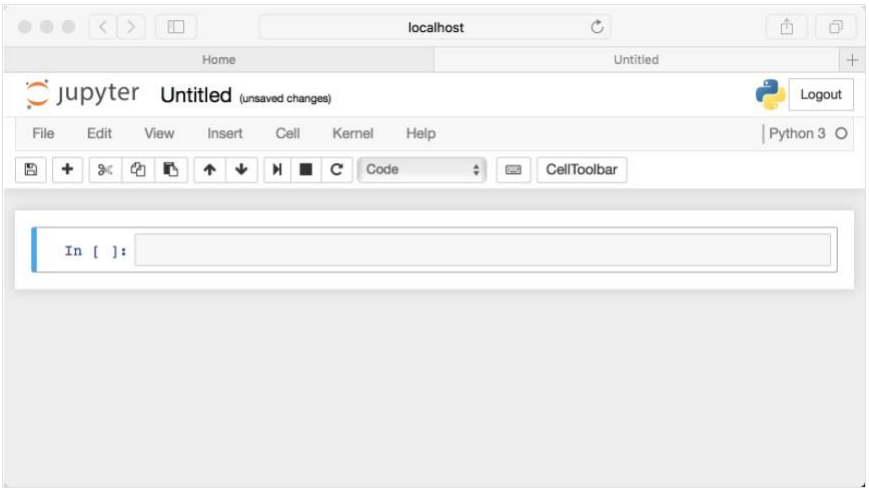
Note that right now you are not actually running a Notebook, but instead you are just running the Notebook server. Let's actually create a Notebook now!

Creating a Notebook

Now that you know how to start a Notebook server, you should probably learn how to create an actual Notebook document.

All you need to do is click on the ***New*** button (upper right), and it will open up a list of choices. On my machine, I happen to have Python 2 and Python 3 installed, so I can create a Notebook that uses either of these. For simplicity's sake, let's choose Python 3.

Your web page should now look like this:



6.5 Python Libraries:

Python Libraries are a set of useful functions that eliminate the need for writing codes from scratch.There are over 137,000 python libraries present today.Python libraries play a vital role in developing machine learning, data science, data visualization, image and data manipulation applications and more. Let us start witha brief introduction to Python Programming Language and then directly dive

into the most popular Python libraries.The probability that you must have heard of 'Python' is outright. Guido Van Rossum's brainchild – Python, which dates back to the '80s has become an avid game changer. It is one of the most popular coding languages today and is widely used for a gamut of applications.

What is a Library?

A library is a collection of pre-combined codes that can be used iteratively to reduce the time required to code. They are particularly useful for accessing the pre-written frequently used codes, instead of writing them from scratch every single time. Similar to the physical libraries, these are a collection of reusable resources, which means every library has a root source. This is the foundation behind the numerous open-source libraries available in Python.

Figure: Python Libraries

List of Libraries Used:

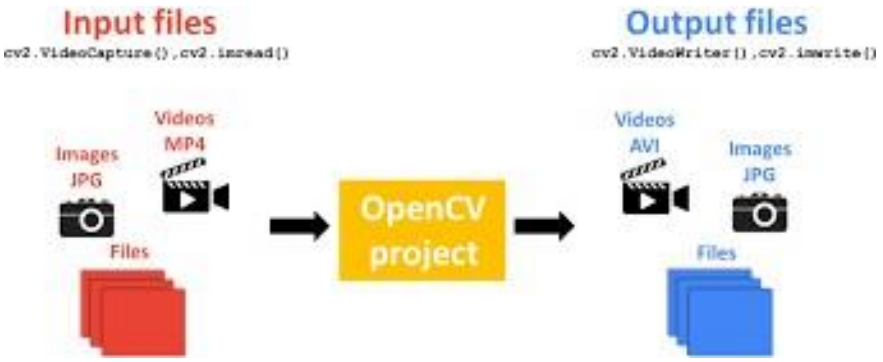
1. **Face Recognition:** Recognize and manipulate faces from Python or from the command line withthe world's simplest face recognition library.

```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_locations = face_recognition.face_locations(image)
```

2. Python OpenCV | cv2.imread() method:

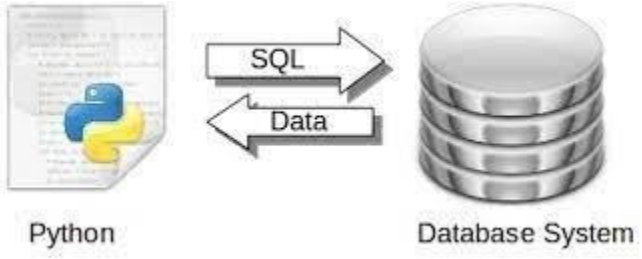
OpenCV-Python is a library of Python bindings designed to solve computer vision problems.cv2.imread() method loads an image from the specified file. If the image

cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.



3. MYSQL.CONNECTOR:

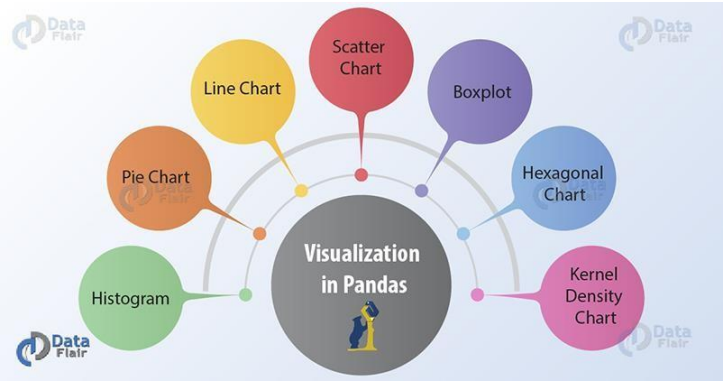
MySQL Connector/Python enables **Python** programs to access **MySQL** databases, using an API that is compliant with the **Python** Database API Specification v2. 0 (PEP 249). **MySQL Connector/Python** includes support for: Almost all features provided by **MySQL** Server up to and including **MySQL** Server version 8.0.

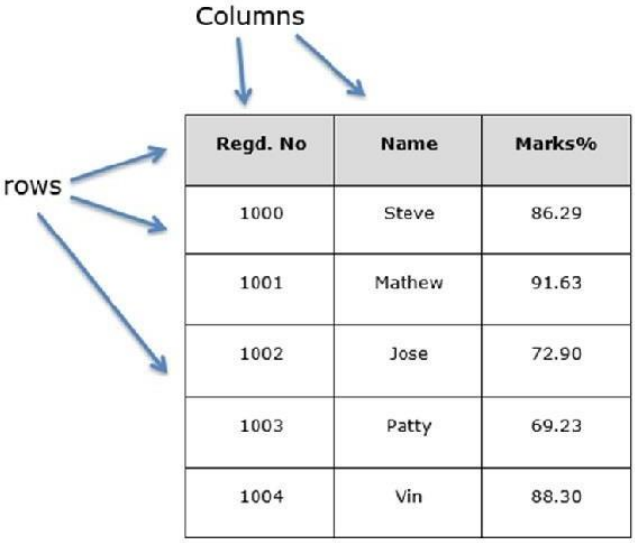


4. PANDAS:

pandas is a **Python package** providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in **Python**.

Pandas is an open source **Python package** that provides numerous tools for data analysis. The **package** comes with several data structures that can be used for many different data manipulation tasks.





Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

5.PYTTSX3:

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.

```
engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
```

6. MatPlotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

- **Create:**

  Develop publication quality plots with just a few lines of code

Use interactive figures that can zoom, pan, update...

- **Customize:**

  Take full control of line styles, font properties, axes properties...

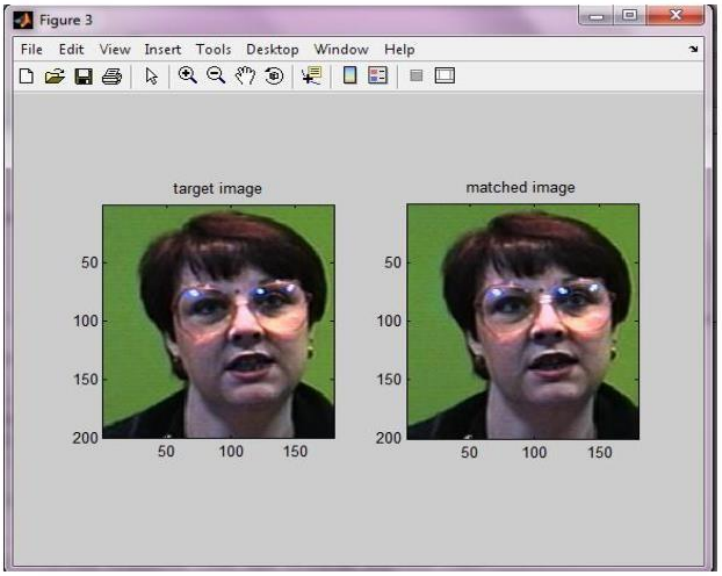
Export and embed to a number of file formats and interactive environments

- **Extend:**

  Explore tailored functionality provided by third party packages

Learn more about Matplotlib through the many external learning resources

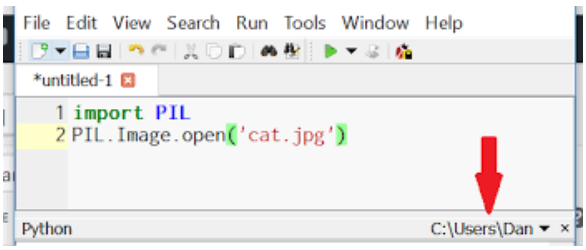
Figure 3: result of face recognition system.



his simulation shows that a facial image having different

7.From PIL import image:

To load the image, we simply import the image module from the pillow and call the Image. open(), passing the image filename. Instead of calling the Pillow module, we will call the PIL module as to make it backward compatible with an older module called Python Imaging Library (PIL).



8. Openpyxlload_workbook():

1. To use load workbook function you should know the name of Excel file you want to work on.

2. You should know the path of file.

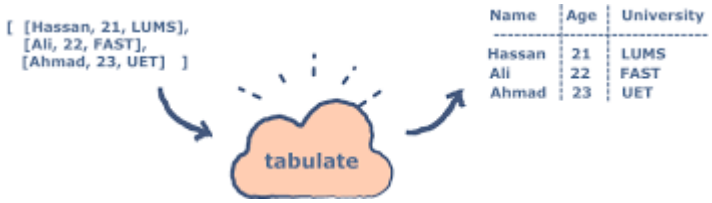
3. Use load_workbook(path+name) to create a reference to that particular Excel file.

4. Use the reference to read, write, create a new sheet, delete a sheet etc.

	A	B	C	D	E
1	STUDENT 'S NAME	COURSE	BRANCH	SEMESTER	
2	ANKIT RAI	B.TECH	CSE	4	
3	RAHUL RAI	M.TECH	CSE	2	
4	PRIYA RAI	MBA	HR	3	
5	AISHWARYA	B.TECH	CSE	4	
6	HARSHITA JAISWAL	B.TECH	BIOTECH	5	
7					
8					
9					
10					
11					
12					
13					
14					

9. From tabulate import tabulate:

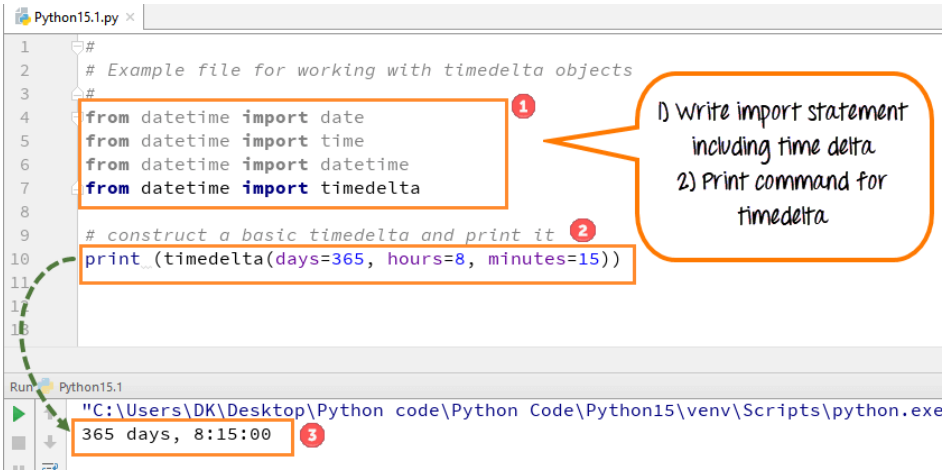
- Printing small tables without hassle: just one function call, formatting is guided by the data itself.
- Authoring tabular data for lightweight plain-text markup: multiple output formats suitable for further editing or transformation.
- Readable presentation of mixed textual and numeric data: smart column alignment, configurable number formatting, alignment by a decimal point.



10. From datetime import datetime, timedelta:

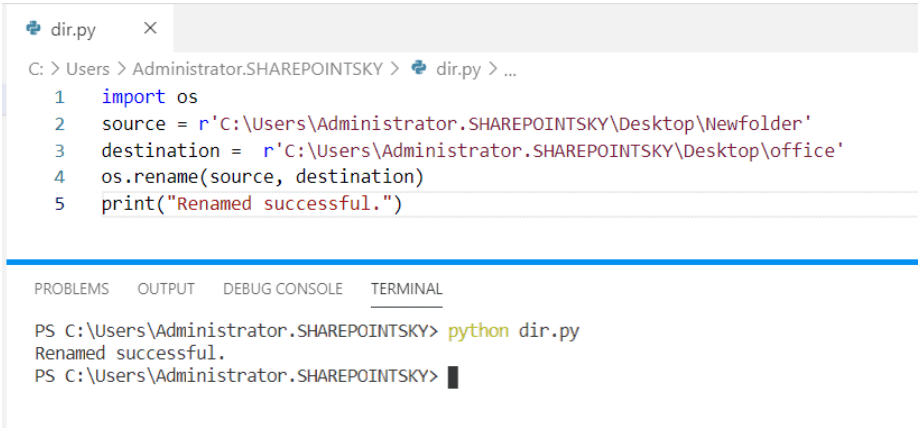
A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects.

- Python timedelta() function is present under datetime library which is generally used for calculating differences in dates and also can be used for date manipulations in Python. It is one of the easiest ways to perform date manipulations.
- The timedelta is the class in datetime module that represents duration. The delta ... or time instances. By using timedelta, you may estimate the time for future and past from datetime import datetime, timedelta.



11. import os:

The os module is a part of the standard library, or stdlib, within Python 3. This means that it comes with your Python installation, but you still must import it. All of the following code assumes you have os imported. Because it is not a built-in function, you must always import it.



SQL:-

SQL(Structured Query Language) is a standard language for accessing and manipulating databases.

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

A face recognition task requires to store multidimensional array and make calculations over it. This task is not mostly matching the capabilities of relational databases and SQL. However, RDBMS comes with elegant, neat and well organized structures. In this post, we will mention how to use a relational database in a face recognition pipeline.

Creating Tables:-

In this project we created two tables

1. Parent Table
2. Child Table

1.Parent Table:- It is used to store the personal information like ID, Name, Department, Gender, Mail id, Mobile number, Location……

	id	Name	branch	mail_id	Gender	mobile_number	location	Transport
0	491	Gokul vamsi	ECE-B	gokulvamsi@gmail.com	None	9676088778	rajahmundary	None
1	492	k.Eswar rao	ECE-B	eswarao@gmail.com	None	9398434537	srikakulam	None
2	493	M.Akhila	ECE-B	akhila@gmail.com	None	9652145866	jangareddygudam	None
3	494	M.Vinitha	ECE-B	mangamvinitha@gmail.com	FeMale	6309428906	kakinada	Bus
4	495	M.vamsi	ECE-B	mvsuryavamsi@gmail.com	Male	8555061635	peddapuram	own
5	496	M.Sati reddy	ECE-B	sathireddy@gmail.com	None	9121364899	mamidada	None
6	497	P.Bharathwaj	ECE-B	bhatarhwaj@gmail.com	None	9347960906	Ongle	None
7	498	N.nookaraj	ECE-B	nookaraj@gmail.com	Male	6300563083	Gumlladhodi	Bus
8	499	chathurya	ECE-B	chathurya@gmail.com	FeMale	8830767406	rajahmundry	Bus

Syntax:

```
Create Table parent (Id int, Name vchar(200), Branch vchar(200), Gender
Vchar(200), Mail_idvchar(200), mobile_number int, location vchar(200), Transport
vchar(200);
```

2.Child Table:- Child table stores the performance data of a person in table format containing data present in rows and columns.

Syntax:

```
Create Table Child (sem int, percentageint, Attendanceint, BLint;
```

	id	sem	percentage	attendance	BL
0	491	6th	45	65	3
1	492	6th	56	71	2
2	493	6th	61	71	0
3	494	6th	75	76	0
4	495	6th	55	79	1
5	496	6th	68	59	1
6	497	6th	72	63	0
7	498	6th	59	66	0
8	499	6th	63	78	0

What is Natural Join in SQL?

We have already learned that an EQUI JOIN performs a JOIN against equality or matching column(s) values of the associated tables and an equal sign (=) is used as comparison operator in the where clause to refer equality.

The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with the same name of associated tables will appear once only.

Natural Join: Guidelines

- The associated tables have one or more pairs of identically named columns.
- The columns must be the same data type.
- Don’t use ON clause in a natural join.

Syntax:

```
"select * from  parent natural join child where name="+"\'"+name+"\'"
```

id	Name	branch	mail_id	Gender	mobile_number	location	Transport	sem	percentage	attendance	BL
491	Gokul vamsi	ECE-B	gokulvamsi@gmail.com	None	9676088778	rajahmundary	None	6th	45	65	3
492	k.Eswar rao	ECE-B	eswarao@gmail.com	None	9398434537	srikakulam	None	6th	56	71	2
493	M.Akhila	ECE-B	akhila@gmail.com	None	9652145866	jangareddygudam	None	6th	61	71	0
494	M.Vinitha	ECE-B	mangamvinitha@gmail.com	FeMale	6309428906	kakinada	Bus	6th	75	76	0
495	M.vamsi	ECE-B	mvsuryavamsi@gmail.com	Male	8555061635	peddapuram	own	6th	55	79	1
496	M.Sati reddy	ECE-B	sathireddy@gmail.com	None	9121364899	mamidada	None	6th	68	59	1
497	P.Bharathwaj	ECE-B	bhatarhwaj@gmail.com	None	9347960906	Ongle	None	6th	72	63	0
498	N.nookaraj	ECE-B	nookaraj@gmail.com	Male	6300563083	Gumlladhodi	Bus	6th	59	66	0
499	chathurya	ECE-B	chathurya@gmail.com	FeMale	8830767406	rajahmundry	Bus	6th	63	78	0

APPLICATIONS:-

While face recognition has been around in one form or another since the 1960s, recent technological developments have led to a wide proliferation of this technology. This technology is no longer seen as something out of science fiction movies like Minority Report. With the release of the iPhone X, millions of people now literally have face recognition technology in the palms of their hands, protecting their data and personal information. While mobile phone access control might be the most recognizable way face recognition is being used, it is being employed for a wide range of use cases including preventing crime, protecting events and making air travel more convenient.

We've compiled a list of 21 ways that face recognition is currently being used to make the world safer, smarter and more convenient.

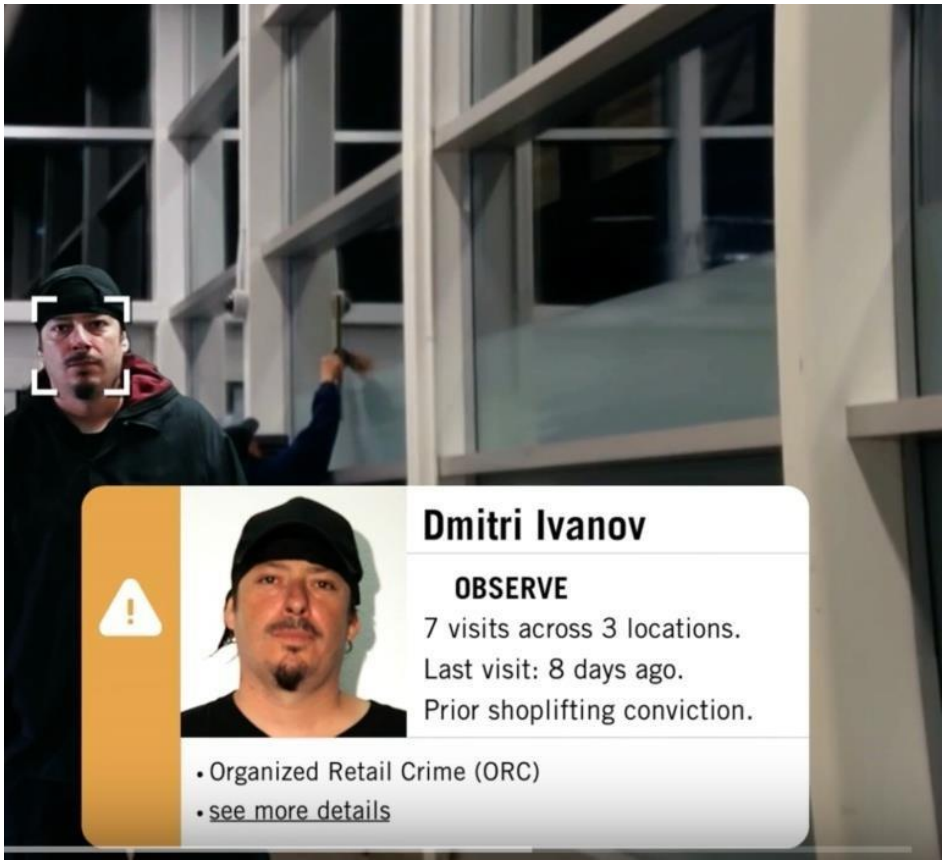


Fig. Capturing criminals

PREVENT RETAIL CRIME

Face recognition is currently being used to instantly identify when known shoplifters, organized retail criminals or people with a history of fraud enter retail establishments. Photographs of individuals can be matched against large databases of criminals so that loss prevention and retail security professionals can be instantly notified when a shopper enters a store that prevents a threat. Face recognition systems are already radically reducing retail crime. According to our data, face recognition reduces external shrink by 34% and, more importantly, reduces violent incidents in retail stores by up to 91%.

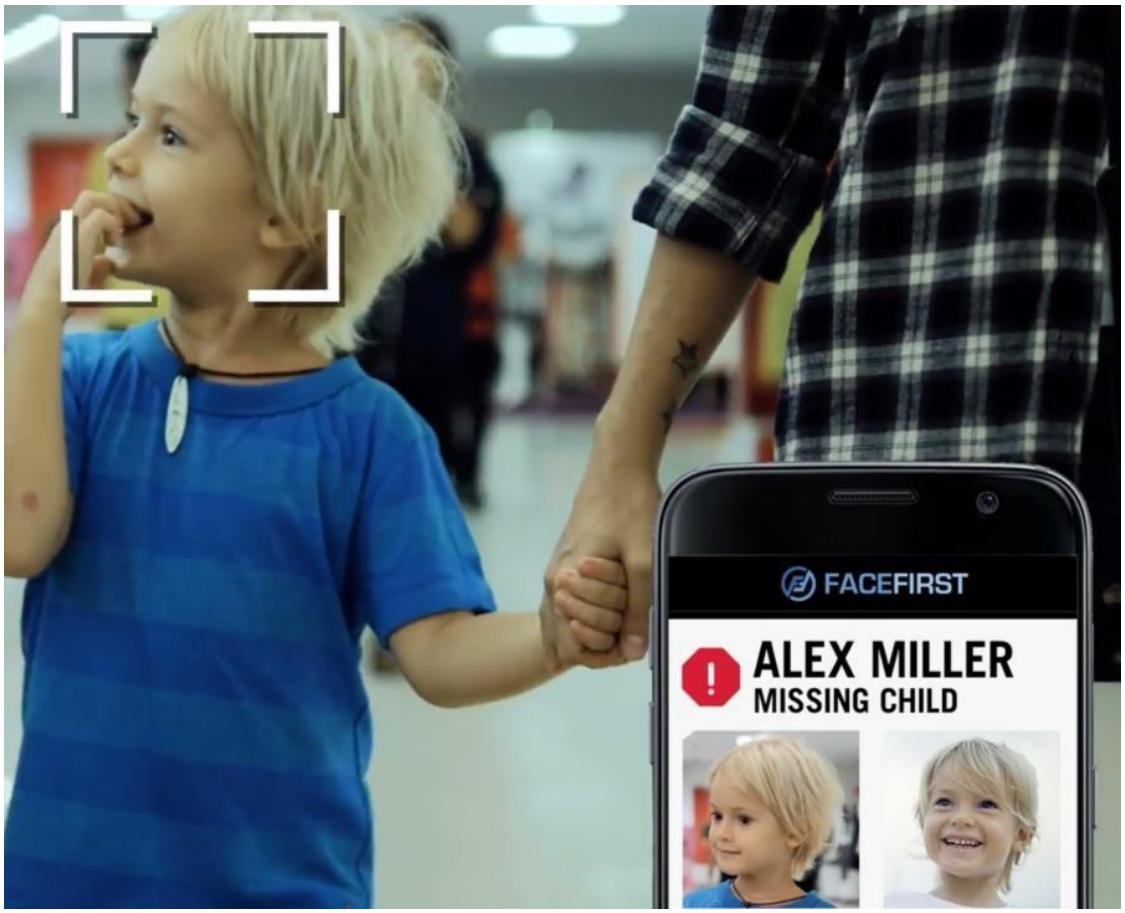


FIG.FINDING MISSING PERSONS

Face recognition can be used to find missing children and victims of human trafficking. As long as missing individuals are added to a database, law enforcement can become alerted as soon as they are recognized by face recognition—be it an airport, retail store or other public space. In fact, 3000 missing children were discovered in just four days using face recognition in India!

PROTECT LAW ENFORCEMENT

Fig.capturing suspects

Mobile face recognition apps, like the one offered by FaceFirst, are already helping police officers by helping them instantly identify individuals in the field from a safe distance. This can help by giving them contextual data that tells them who they are dealing with and whether they need to proceed with caution. As an example, if a police officer pulls over a wanted murderer at a routine traffic stop, the officer would instantly know that the suspect may be armed and dangerous, and could call for reinforcement.

PROTECT SCHOOLS FROM THREATS:-

Face recognition surveillance systems can instantly identify when expelled students, dangerous parents, drug dealers or other individuals that pose a

Fig. unauthorised person in orginasations,schools

threat to school safety enter school grounds. By alerting school security guards in real time, face recognition can reduce the risk of violent acts.

TRACK SCHOOL ATTENDANCE:-

In addition to making schools safer, face recognition has the potential to track students' attendance. Traditionally, attendance sheets can allow students to sign another student, who is ditching class, in. But China is already using face recognition to ensure students aren't skipping class. Tablets are being used to

scan students' faces and match their photos against a database to validate their identities.

CONTROL ACCESS TO SENSITIVE AREAS:-

Face recognition can work as a means of access control to ensure that only authorized individuals get into facilities like labs, boardrooms, bank vaults, training centers for athletes and other sensitive locations.

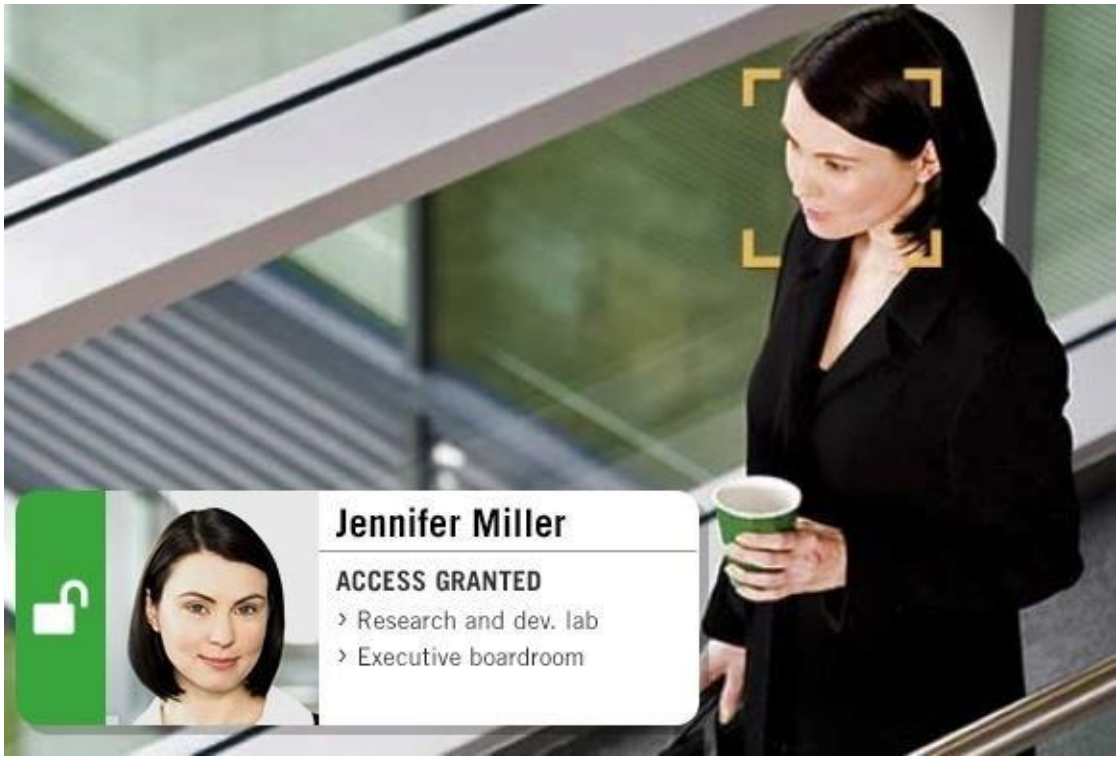


Fig. capturing person within the orginazation

ADVANTAGES:

It takes barely second to recognize your face and create attendance journal in the payroll system, saving precious man-hours

ACCURACY

Automatically scans multiple employees, with 99.8% accuracy in face detection. Faces once recognised are accurately enrolled into the payroll system

OPTIMIZE RESOURCES

Makes data available in real-time and synchronizes the data without a time-lag. This real-time monitoring helps to optimize workforce and other

INCREASED PRODUCTIVITY

Saving time will lead to an increase in productivity, resulting in reduction of cost (optimal utilization of resources) and an increase in revenue

WELL-INTEGRATED

Readymade integration with Ramco's HR and Payroll modules and other modules as well unlike other time marking systems that require efforts to develop integration components

HYGIENIC

The system ensures that there are no concerns regarding hygiene and cleanliness unlike biometric systems which has devices requiring physical contact.

Towards a Touchless Attendance System:

Move over finger based biometrics and unhygienic plastic based access cards - mark attendance in the most hygienic and efficient way possible

EXTENDING FEATURES OF FACE RECOGNITION

Intelligent Features of Facial Recognition

LIVENESS DETECTION

Ensures no one uses a photograph or a video to feign attendance, preventing fraud attendance entries

GEO-FENCING

Tracks people's movement within specific work areas; useful for offices with restricted areas which permits only select employees

OFFLINE CAPABILITIES

Enables businesses with poor internet access to register attendance offline, and later allows bulk upload of data when there's a strong network

CHAPTER -7

Results and Discussion

Code:-

```
import face_recognition
import cv2
import mysql.connector
import pandas as pd
import pyttsx3
from PIL import Image
from openpyxl import load_workbook
import matplotlib.pyplot as plt
from tabulate import tabulate
from datetime import datetime,timedelta
import os


mydb = mysql.connector.connect(
host = "localhost",
user="vamsi",
password="V@msi495",
database ="python"
)



#for speech perpose
engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
engine.setProperty('rate', 160)
engine.setProperty('volume',1.0)



# capturing image from web cam and verifying the image....
video_capture =cv2.VideoCapture(0)
vamsi_image = face_recognition.load_image_file(r"C:\\vamsi.jpeg")
vamsi_face_encoding =
face_recognition.face_encodings(vamsi_image)[0]
```

```
vinitha_image =
face_recognition.load_image_file(r"C:\Users\mvsur\Downloads\vinitha2
.jpeg")
vinitha_face_encoding =
face_recognition.face_encodings(vinitha_image)[0]

chathurya_image =
face_recognition.load_image_file(r"C:\Users\mvsur\Downloads\chatury
a.jpeg")
chathurya_face_encoding =
face_recognition.face_encodings(chathurya_image)[0]

nook_image = face_recognition.load_image_file(r"C:\nooks.jpeg")
nook_face_encoding =
face_recognition.face_encodings(nook_image)[0]


# Create arrays of known face encodings and their names
pics=[
    vamsi_image,
    vinitha_image,
    chathurya_image,
    nook_image
]
known_face_encodings = [
    vamsi_face_encoding,
    vinitha_face_encoding,
    chathurya_face_encoding,
    nook_face_encoding
]
known_face_names = [
    "M.vamsi",
    "M.vinitha",
    "chathurya",
    "N.nookaraj"

]



def details():
    while True:
        ret, frame = video_capture.read()
        face_locations = face_recognition.face_locations(frame)
        face_encodings = face_recognition.face_encodings(frame,
face_locations)
        name = "unknown"
        for (top, right,bottom,left),face_encodings in zip(face_locations,
face_encodings):
```

```
            matches =
face_recognition.compare_faces(known_face_encodings,
face_encodings)


            if True in matches:
                first_match_index = matches.index(True)
                name = known_face_names[first_match_index]
                pic = pics[first_match_index]
            cv2.rectangle(frame,(left, bottom - 35), (right, bottom),(0,255,0),
cv2.FILLED)
            font= cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame,name, (left +6,bottom -6), font,
1.0,(255,255,255),1)
        cv2.imshow('recogniting',frame )

        if cv2.waitKey(1) & 0xFF ==   ord('q'):
            break

        if name in known_face_names:
            #from here getting data from the sql data base
            #print(name)
            mycursor = mydb.cursor()
            mycursor.execute("select * from  parent natural join child where
name ="+"\'"+name+"\'")
            myresult = mycursor.fetchall()
            field = [i[0] for i in mycursor.description]
            df1=pd.DataFrame(myresult,columns=field)
            return name,df1,pic
        else:
            engine.say('unkown person, data not found in data base')
            engine.runAndWait()
            return details()

def display_details():
        name,df1,pic,=details()
        plt.imshow(pic)
        b=df1
        engine.say('Hi'+name+'here is your details')

        k=[]
        l=[]
        for i in b:
            k.append(i)
            for j in df1[i].values:
                l.append(j)
        dict = {'Name':k,'values':l}
        df = pd.DataFrame(dict)
```

```
        # displaying the DataFrame
        print(tabulate(df,tablefmt = 'fancy_grid'))


        #break
def attendance():
    name,df2,pic=details()
    file_dt=datetime.strftime(datetime.now()+timedelta(0),'%d-%m-%Y')
    login_time=datetime.now().strftime("%H:%M:%S")
    for i in df2.id.values:
        a=i
        for j in df2.Name.values:
            b=j
    pat=r'C:\Users\mvsur\Documents\final_year_attendance.xlsx'
    wb= load_workbook(pat)
    t=[a,b,login_time]

    if file_dt in wb.sheetnames:
        ws=wb[file_dt]
        ws.append(t)
        engine.say('Hello'+df2.Name.values+'welcome,have a nice day')
        print("{} is present".format(name))
    else:
        wb.create_sheet(file_dt)
        ws=wb[file_dt]
        ws.append(t)
        engine.say('Hello'+df2.Name.values+'welcome,have a nice day')
        print("{} is present".format(name))




    wb.save(pat)
    #return attendance()




    #return details()

n = int(input('''
            choose option:

                    1.GET PERSON DATA
                    2.ATTENDANCE

            '''))
if n==1:
    display_details()
elif n== 2:
    attendance()
```

```
engine.runAndWait()
video_capture.release()
cv2.destroyAllWindows()
```

Initial input:-

Based on our Requirement we can choose options.

- If we choose option 1(GET PERSON DATA), It will give the detail report of a person data. If data exists in organisation database.
- If we choose option 2(ATTENDANCE), It will work as a attendance recorder along with LOGIN TIME and final report is given by EXCEL SHEET with Day-to-Day report.

Input:-

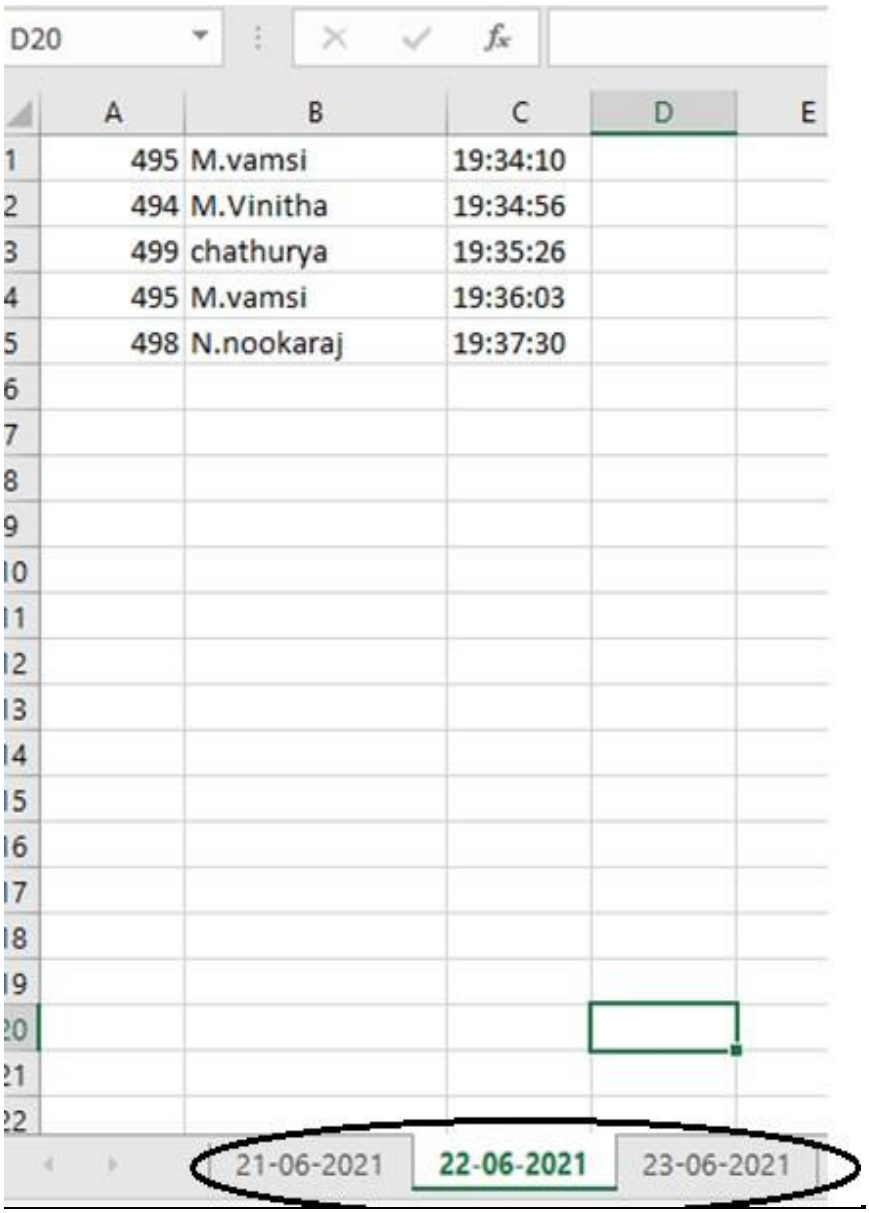
```
choose option:

1.GET PERSON DATA
2.ATTENDANCE
```

Output:- (For option 1):-

0	id	498
1	Name	N.nookaraj
2	branch	ECE-B
3	mail_id	nookaraj@gmail.com
4	Gender	Male
5	mobile_number	6300563083
6	location	Gumlladhodi
7	Transport	Bus
8	sem	6th
9	percentage	59
10	attendance	66
11	BL	0

Output:- (For option 2)



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 eigen faces 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. 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 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. 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.

This system aims to build an effective attendance and detailed report using face recognition techniques. The proposed system will be able to mark the attendance via face Id. It will detect faces via webcam and then recognize the faces. After recognition, it will mark the attendance of the recognized student and update the attendance record. It also used in the seminars where large number of students gathered. And we should provide a number of students attended to a seminar in less time with more accuracy by capturing the facial features of the student. Attendance system helps in increasing the accuracy and speed achieve the high-precision real time attendance to meet the need for automatic classroom evaluation. It is hygienic when compared to the other methods. This method will have higher accuracy in recognition of multiple faces from a single frame with lower response time. And provide a day to day.

REFERENCES

[1] Learning OpenCV –Computer Vision with the OpenCV Library O'Reilly Publication.

[2] Learning OpenCV: Computer Vision with OpenCV Library, Kindle Edition.Gary Bradsk1 and AndrianKehlar

[3] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[4] "KyungnamKim" Face Recognition using Principle Component Analysis"

[5] Codacus: https://youtu.be/1Jz24sVsLE4

[6] G B Huang, H Lee, E L. Miller, "Learning hierarchical representation for Face verification with convolution deep belief networks[C]",Proceedings of International Conference on ComputerVision and Pattern Recognition,pp.223-226,2012.

[7] Computer Vision Papers, http://www.cvpapers.com

[8] Learning OpenCV: Computer Vision with the OpenCV Library 1st Edition, Kindle Edition

[9] OpenCV Homepage http://opencv.willowgarage.com

[10] Recognition Homepage http://www.face-rec.org/algorithms.

[11] Paul Viola, Matthew Jones Conference paper- IEEE Computer Society Conference on Computer Vision and Pattern Recognition."Rapid object detection using a boosted cascade of simple features."

[12] Beymer, D. and Poggio, T. (1995) Face Recognition From One Example View, A.I. Memo No. 1536, C.B.C.L. Paper No. 121. MIT

[13] Brunelli, R. and Poggio, T. (1993), Face Recognition: Features versus Templates. IEEE

[14] Goldstein, A.J., Harmon, L.D., and Lesk, A.B. (1971). Identification of human faces. In Proc. IEEE, Vol. 59, page 748

[15] de Haan, M., Johnson, M.H. and Maurer D. (1998) Recognition of individual faces

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

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

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

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

.[20] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In2009 IEEE 12th International Conference on Computer Vision, pages 32–39, Sept 2009

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

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

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

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

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

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

[27] Meng Xiao He and Helen Yang. Microarray dimension reduction, 2009.