**CHAPTER-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.

**Face Recognition:**

**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 1.2)
2. **Photometric Stereo:** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006) (Figure 1.1)

**Popular recognition algorithms include:**

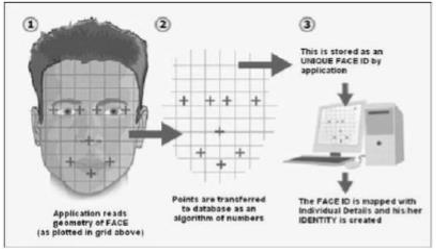
1. Principal Component Analysis using Eigenfaces (PCA),

2. Linear Discriminate Analysis,

3. Elastic Bunch Graph Matching using the Fisherface algorithm,



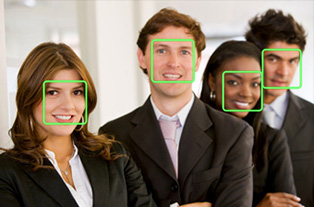
**Figure 1.1:** Photometric Stereo Image



**Figure 1.2:** Geometric Facial Recognition

**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).



**Figure 1.3:** Face Detection Process

**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 no faces by training on these examples. We use both our implementation of the neural network and the OpenCV neural network toolbox for this task. Different network configurations are experimented with to optimize the results.

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

**Objective:**

* Trying to find a face within a large database of faces. In this approach the system returns a possible list of faces from the database. The most useful applications contain crowd surveillance, video content indexing, personal identification (example: drivers license), mug shots matching, etc.
* Real time face recognition: Here, face recognition is used to identify a person on the spot and grant access to a building or a compound, thus avoiding security hassles. In this case the face is compared against a multiple training samples of a person.

**Technology:**

**Python:**

The technology we use for this project is python. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

**OpenCV Library:**

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python. OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn’t one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers. For something like a face, you might have 6,000 or more classifiers, all of which must match for a face to be detected (within error limits, of course).

**Methodology Used:**

We have used Open CV for facial recognition system. OpenCV has three built-in face recognizers and thanks to its clean coding, we can use any of them just by changing a single line of code. Here are the names of those face recognizers and their OpenCV calls:

* **EigenFaces** – cv2.face.createEigenFaceRecognizer()
* **FisherFaces** – cv2.face.createFisherFaceRecognizer()
* **LocalBinary Patterns Histograms (LBPH)** –cv2.face.createLBPHFaceRecognizer()

In this project, we have used Local Binary Patterns Histograms for facial recognition.



**Figure 1.4:** OpenCV Facial landmark Detection

**CHAPTER-2**

**SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)**

### **Introduction**

### **Purpose**

The purpose of this System Requirement Specification document is describing the security system which is called Bio-metric identification based on Face. This system aims to provide a security system which holds a personal information keep in safe and decrease the rate of information theft against who want to steal your private information. This document includes detailed information about requirements of the project. Overall, this document is used for how users interact with the system and understand how the mechanism works at back-end without any problems and explains how concerns of the stakeholders are met.

### **Scope of Project**

Most of the people use a private computer to do their jobs in the company and they may need to hide information in documents which relevant to work. Some information can be public and this files that are not important, if they are seized by someone else, but some files need a special protection system which is in the high-level secret status because people are wasting their time for hours on end and some hacker can steal their information from victim’s computer easily without any protection system and worst of all, people are unprepared for this situation. The application to be improved is Recognition of Human Face Patterns for Biometric Identification. This project involves developing a face detection system in order to verify the uniqueness of the human face by detecting the face pattern from the image.

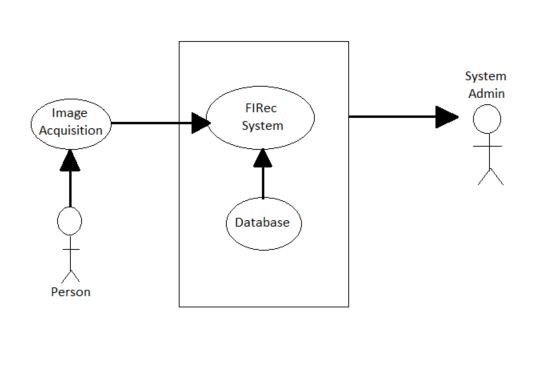
### **Glossary:**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | A person whose Face is to be recognized |
| Admin | A person who handles the application and enrolls a person image in the database |
| Database | Collection of all the information about the face and data of a person |
| Face Recognition | Control system for using human face for login to the system |
| Features | Result of information of the matrix which derives from grayscale image |

### **Overall Description**

### **System Environment**

This application is Face Recognition System as seems as Fig 1. Firstly, A person’s image should be provided using the camera. Then, camera transfers it to the Face Recognition System. Next, an image is preprocessed for features of face. After that, they are compared with an image in the system database and matched. The System Admin can do add and delete operations in the database, can set appropriate match rate of accepting user and handle whole application.



**Figure 2.1:** Process of Face Recognition

### **Development methodology**

While developing the project, we have decided to use Scrum which is an agile software development methodology. Scrum is part of the agile movement in a scrum, it has a sprint which includes work to do in the project. It takes a while almost between 2 and 4 weeks.

### **Functional Requirements Specification**

In this section, use cases are outlined for every single actor. System admin is the main actor and just one use case belongs to the user. Functional Requirements of Requirement Specification and this section are similar to each other.

### **User Interface Specification**

The framework holder is anticipated to be able to utilize button, pull-down menus, and comparative devices. A window with graphical client interface will be utilized. To begin with, of all we have chosen to store the highlights of face shape of the pictures of a face of people and utilize those highlights for acknowledgment. In case the input image's highlights coordinate up with the already put away highlights in the database, at that point a message will be shown illuminating an effective acknowledgment along with the subtle elements of the comparing individual.

### **Non-Functional Requirements**

In the Face Recognition system, the input is an image. The system will be implemented in Python. The user should not move during face detection in order to obtain high quality results. In addition, the environment should not be too bright and dark while detecting the face. Oracle database will be used for store the features which include a User’s face pattern and make a comparison between active employee’s features with features for finding the current employee is exist in the database. Operating system will execute the system.

### **Performance the System Requirements**

The system will accept or reject user in 4 seconds, after scanning.

### **Requirements Specifications**

### **External Interface Requirements:**

### **User Interfaces:**

The user interface will run on Windows.

### **Hardware Interfaces:**

The Face Recognition system require a camera. The camera requires necessary driver installed within the operating system.

### **Software Interfaces:**

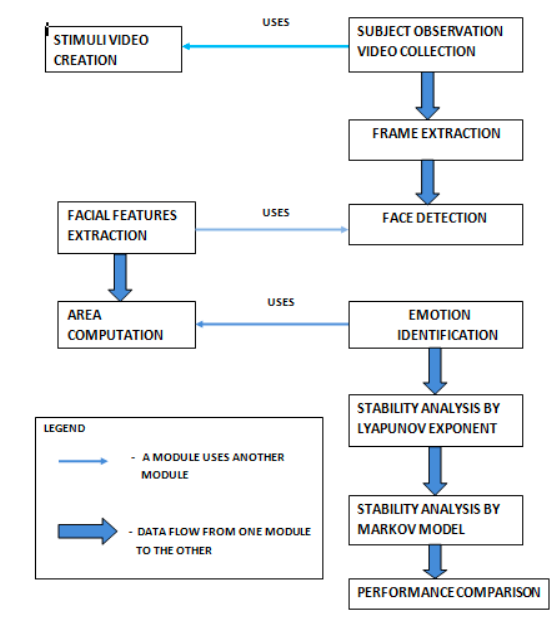
There are no external software interface requirements.

### **Communication Interfaces:**

There are no external communications interface requirements

### **Functional Requirements**

**Use Case Diagrams:**



**Figure 2.2:** Diagram of Face Recognition

### **Detailed Non-Functional Requirements**

### **Security:**

The computer that runs the program will have its own security. Only the System Admin will log in to the system with his/her username and password. The person whose face are recognized will access to view the output.

### **Maintainability:**

As a tool to obtain the ease of maintainability UML will be used in the development process.

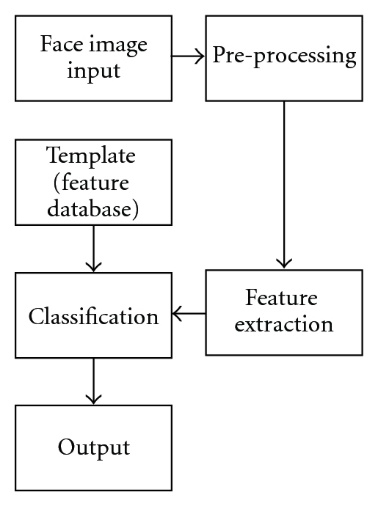
**Portability:**

To ensure portability, the application will be developed in Python language and OpenCV Library.

**System Evolution**

The Face Recognition application are developed in Python. We are using Oracle as a database for storing features of employees. In the future, we can robust the algorithm to provide more accurate and consistent match-making recognition system. Also, we will reduce face recognition time under the 4 sec. Face Recognition System is available on the laptop which has an internal camera. Also, we are thinking to use Face Recognition System at bank ATMs and immigration system terminals for avoiding information theft and reducing crime rate for public safety.

**Face Recognition Process**



**Figure 2.3:** Flow Process of Face Recognition

**Performance Requirements**

Camera’s visual must run smoothly without any error and delay more than 4sec to get the image of the user. This requirement is depended on many aspects of the user pc. Minimum requirements for running Face Recognition System are:

|  |  |
| --- | --- |
| **Hardware and Software** | **Type** |
| **Processor** | Intel Core Processor or better performance |
| **Primary Memory** | 1 GB or more |
| **Secondary Memory** | 3 GB or more |
| **Graphics** | 800 MB or more |
| **Printer** | Not Required |
| **Operating System** | Windows and Linux Operating System |

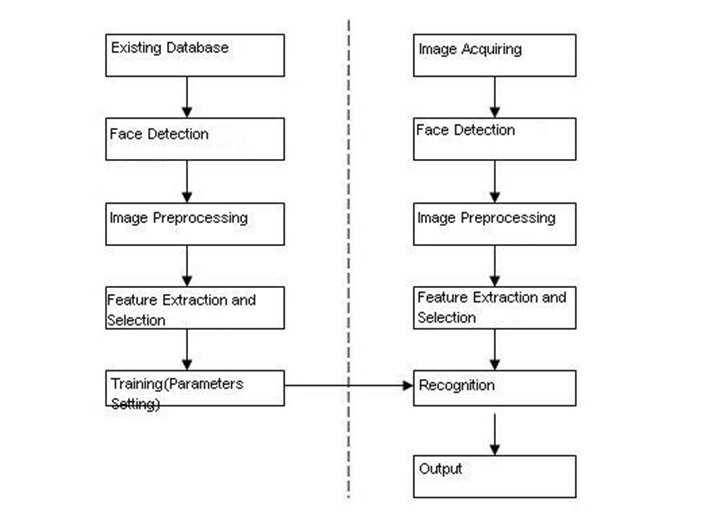
**Table 2.1:** Hardware and Software Requirements

**CHAPTER 3**

**FLOW DIAGRAMS**

**The Use Case Diagram for the Face Recognition System:**

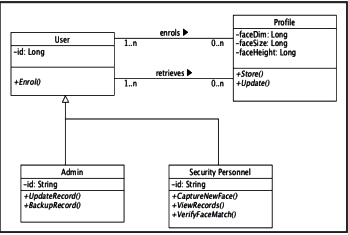
In order to realize the model, we employed the Unified Modeling Language (UML). It uses diagrams to document an object-based decomposition of systems showing the interaction between these objects and the dynamics of these objects. UML aims to provide a common vocabulary of object-based terms and diagramming techniques that is rich enough to model any system development project from analysis to design. For our model, we make use of use case diagram, activity diagram class diagram.



**Figure 3.1:** Use Case Diagram of Face Recognition

Shows the use case diagram of the proposed face recognition system with two actors - the administrator as well as the security personnel. The administrator of the system has full access to all the system features and will often be in an office setting while the security personnel will mostly be on the field and so has access to just the crucial functions required on the field. These includes ability to enrollee and capture a new face; view existing records and try to match a face with existing records activity diagram for verification, identification and matching of new faces to existing ones.

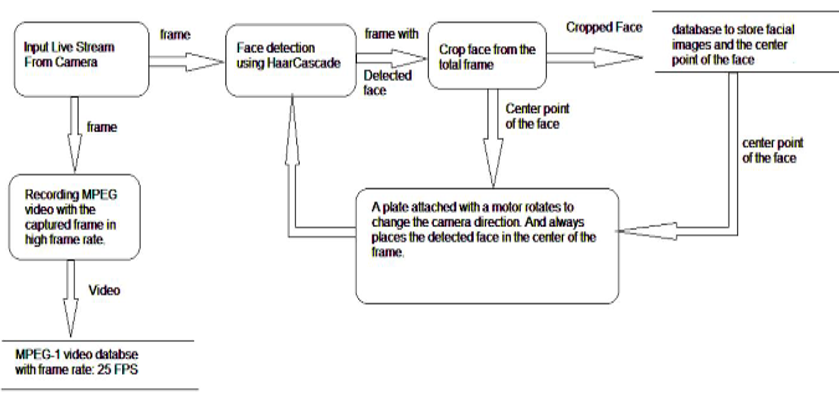
**The Class Diagram of the Face Recognition System:**



**Figure 3.2:** Class Diagram of the Face Recognition

**Data Flow Diagram:**

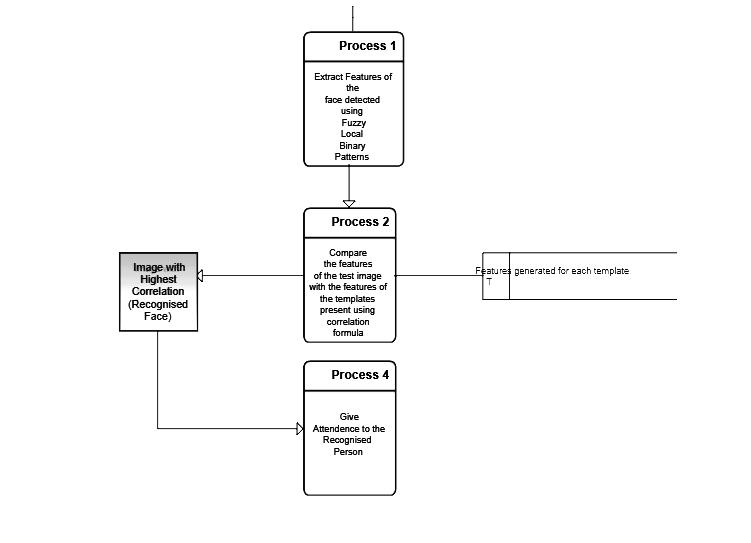
**Level 0 DFD for Facial Recognition System:**



**Figure 3.3:** Level 0 DFD of Face Recognition

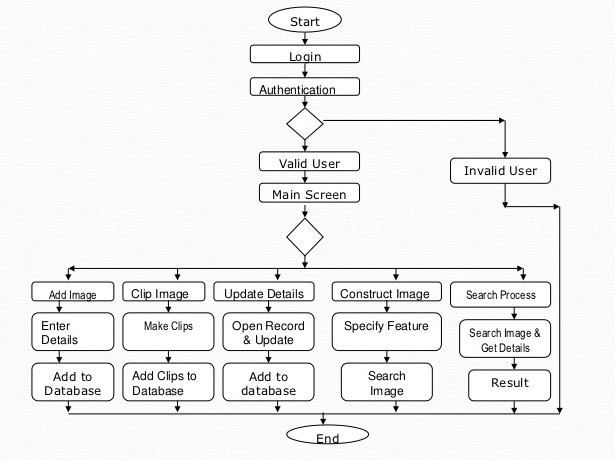
It's easy to understand the flow of data through systems with the right **data flow diagram** software. This guide provides everything you need to know about **data flow diagrams**, including definitions, history, and symbols and notations. You'll learn the different levels of a DFD, the difference between a logical and a physical.

**Level 1 DFD for Facial Recognition System:**



**Figure 3.4:** Level 1 DFD of Facial Recognition System

**Level 2 DFD for Facial Recognition System:**



**Figure 3.5:** Level 2 DFD of Facial Recognition System

**CHAPTER 4**

**PROJECT METHODOLOGY**

**Methodology**

Different steps were implemented during the preparation of the projects. The following methods were employed.

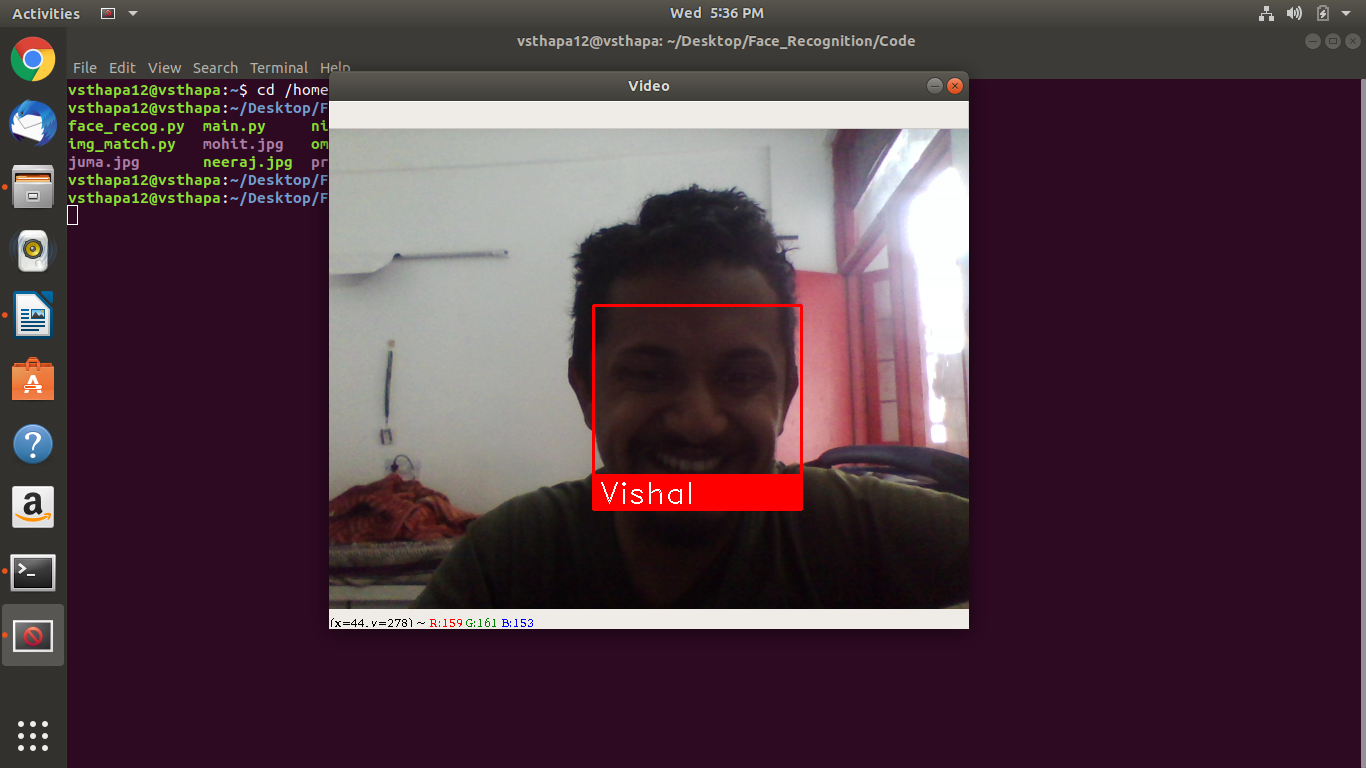
* **Sample survey:** - In this a brief research was made related on the project.
* **Information:** - Required information was collected from the different source such as web and different sources.
* **Opinion collection:** - Opinion related to the project was collected from different source such as friends and teachers.
* **In depth study:** - finally in depth study was made related to the project. Different way of implementing the found resource was studied.
* **Programming:** - After all of the above programming is done.
* **Testing and maintenance:** - after programming testing and maintenance was done.

I followed SDLC (System Development Life Cycle) for project development phases. Systems Development Life Cycle (SDLC) or sometimes just (SLC) is defined by the U.S. Department of Justice (DoJ) as a software development process, although it is also a distinct process independent of software or other Information Technology considerations. It is used by a systems analyst to develop an information system, including requirements, validation, training, and user ownership through investigation, analysis, design, implementation, and maintenance.

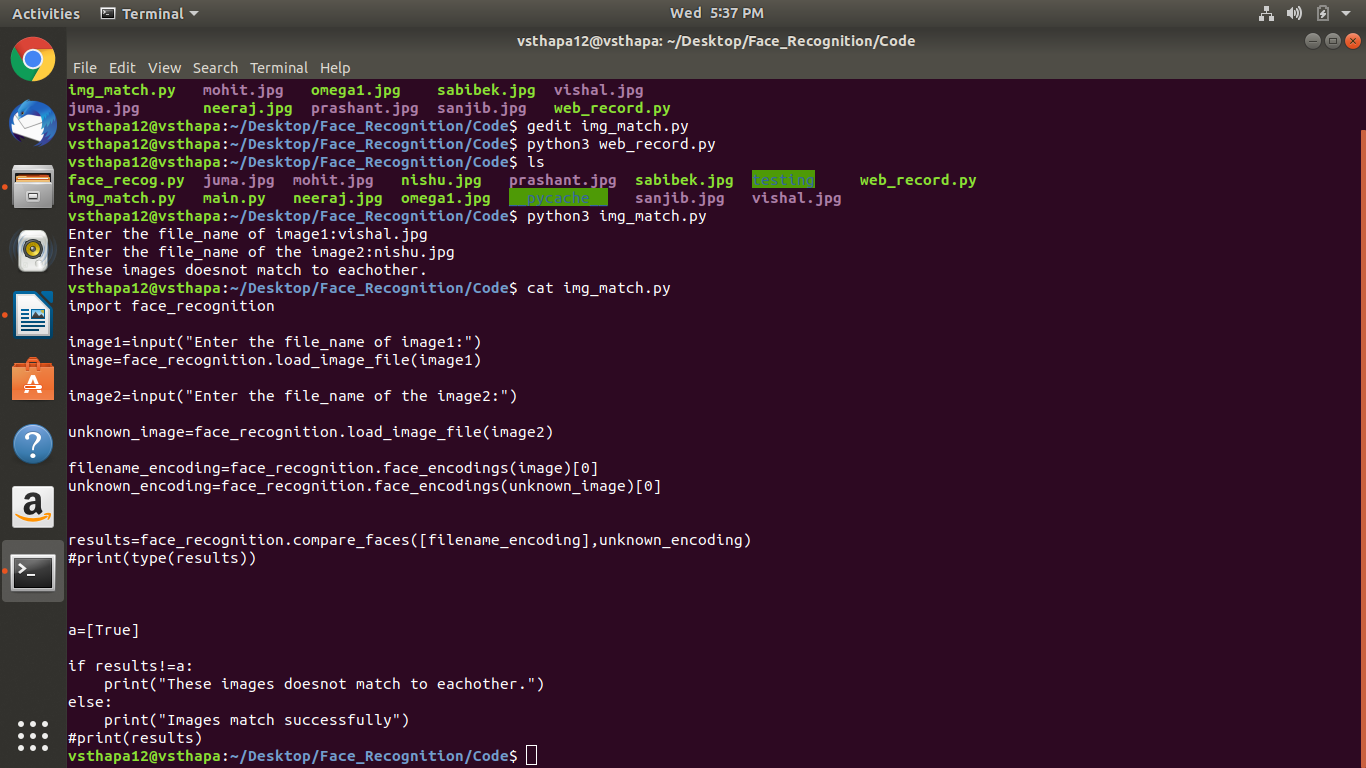
1. **Terms Of Reference:** the management will decide what capabilities and objectives they wish the new system to incorporate
2. **Feasibility Study:** asks whether the managements' concept of their desired new system is actually an achievable, realistic goal, in-terms of money, time and end result difference to the original system. Often, it may be decided to simply update an existing system, rather than to completely replace one;
3. **Fact Finding and Recording**: how is the current system used? Often questionnaires are used here, but also just monitoring (watching) the staff to see how they work is better, as people will often be reluctant to be entirely honest through embarrassment about the parts of the existing system they have trouble with and find difficult if merely asked;
4. **Analysis:** free from any cost or unrealistic constraints, this stage lets minds run wild as 'wonder systems' can be thought-up, though all must incorporate everything asked for by the management in the Terms Of Reference section;
5. **Design**: designers will produce one or more 'models' of what they see a system eventually looking like, with ideas from the analysis section either used or discarded. A document will be produced with a description of the system, but nothing is specific they might say 'touchscreen' or 'GUI operating system', but not mention any specific brands;
6. **System Specification:** having generically decided on which software packages to use and hardware to incorporate, you now have to be very specific, choosing exact models, brands and suppliers for each software application and hardware device;
7. **Implementation and Review:** set-up and install the new system (including writing any custom (bespoke) code required), train staff to use it and then monitor how it operates for initial problems, and then regularly maintain thereafter.

**CHAPTER 5**

**SCREENSHOTS**



**Figure 5.1:** Showing Detected Face with name



**Figure 5.2:** Face Matching Process

**CHAPTER 6**

**CONCLUSION & FUTURE SCOPE**

**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 only reason for this was the face recognition subsystem did not display even a slight degree of invariance to scale, rotation or shift errors of the segmented face image. However, if some sort of further processing, such as an eye detection technique, was implemented to further normalize the segmented face image, performance will increase to levels comparable to the manual face detection and recognition system. Implementing an eye detection technique would be a minor extension to the implemented system and would not require a great deal of additional research. All other implemented systems displayed commendable results and reflect well on the deformable template and Principal Component Analysis strategies. The most suitable real-world applications for face detection and recognition systems are for mugshot matching and surveillance. There are better techniques such as iris or retina recognition and face recognition using the thermal spectrum for user access and user verification applications since these need a very high degree of accuracy. The real-time automated pose invariant face detection and recognition system proposed in chapter seven would be ideal for crowd surveillance applications. If such a system were widely implemented it’s 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.. The automated vision systems implemented in this thesis did not even approach the performance, nor were they as robust as a human's innate face recognition system. However, they give an insight into what the future may hold in computer vision.

**Future Scope:**

The use of spherical canonical images allows us to perform matching in the spherical harmonic transform domain, which does not require preliminary alignment of the images. The errors introduced by embedding into an expressional space with some predefined geometry are avoided. In this facial expression recognition setup, end-to-end processing comprises the face surface acquisition and reconstruction, smoothening, sub sampling to approximately 2500 points. Facial surface cropping measurement of large positions of distances between all the points using a parallelized parametric version is utilized. The general experimental evaluation of the face expressional system guarantees better face recognition rates. Having examined techniques to cope with expression variation, in future it may be investigated in more depth about the face classification problem and optimal fusion of color and depth information. Further study can be laid down in the direction of allele of gene matching to the geometric factors of the facial expressions. The genetic property evolution framework for facial expressional system can be studied to suit the requirement of different security models such as criminal detection, governmental confidential security breaches etc.

**REFERENCES**

1. Baron, R. J. (1981). Mechanisms of human facial recognition. International Journal of Man Machine Studies, 15:137-178
2. 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
3. Bichsel, M. (1991). Strategies of Robust Objects Recognition for Automatic Identification of Human Faces. PhD thesis, , Eidgenossischen Technischen Hochschule, Zurich.
4. Brennan, S. E. (1982) The caricature generator. M.S. Thesis. MIT.
5. Brunelli, R. and Poggio, T. (1993), Face Recognition: Features versus Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10):1042-1052
6. Craw, I., Ellis, H., and Lishman, J.R. (1987). Automatic extraction of face features. Pattern Recognition Letters, 5:183-187, February.
7. Deffenbacher K.A., Johanson J., and O'Toole A.J. (1998) Facial ageing, attractiveness, and distinctiveness. Perception. 27(10):1233-1243
8. Dunteman, G.H. (1989) Principal Component Analysis. Sage Publications.
9. Frank, H. and Althoen, S. (1994). Statistics: Concepts and applications. Cambridge University Press. p.110
10. Gauthier, I., Behrmann, M. and Tarr, M. (1999). Can face recognition really be dissociated from object recognition? Journal of Cognitive Neuroscience, in press.
11. Goldstein, A.J., Harmon, L.D., and Lesk, A.B. (1971). Identification of human faces. In Proc. IEEE, Vol. 59, page 748