Bilkent University

Department of Computer Engineering

Senior Project

Who do you resemble?

High Level Design Report

Group Members

Merve Soner

Merve Yurdakul

R. Baturalp Torun

Sedef Özlen

Supervisor: Pinar Duygulu Şahin

Jury Members: Selim Aksoy, H. Altay Güvenir

January 4, 2008

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Projects course CS491.

Table of Contents

1 Introduction

1.1 Purpose of the system

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. Over the last ten years or so, detecting and recognizing human faces automatically is becoming a very important task in many applications, such as security access control systems or content-based indexing video retrieval systems. It is one of the biggest research areas for computer vision and for most successful applications of image analysis and understanding. [1]

One of the ways to do face recognition is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Popular recognition algorithms include eigenface, fisherface, the Hidden Markov model, and the neuronal motivated dynamic link matching. [7] A newly emerging trend, claimed to achieve previously unseen accuracies, is three-dimensional face recognition. Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. Our aim is to find the most accurate working method among these and improve it, by applying it to find the similarities between celebrities.

1.2 Design Goals

The advanced technology on this topic has been trying to meet the requirements of finding desired information easily. However, when the issue is majored on searching based on images, especially faces, current technology cannot fulfill the requirements. [6] Our goal is to select the fastest, most qualified and robust algorithms and develop applications that carry human interaction with computer and visual data processing onto next level. Our primary focus is specifically faces detection and recognition field.

Our system will be functioned by query including specific face. The system performs face detection on the given input. Face recognition process will be performed together with face detection for matching the most similar faces specified by user.

For our project, we will design high-level system for using multi model data for recognizing face having detected the faces with the specific face given in a query. To accomplish the aim, we will use and develop the existing algorithms of face recognition. We will use the existing systems working on face detection and recognition but the main aim is to find the best system to be developed to fulfill the requirements.

1.3 Definitions, acronyms, and abbreviations

Eigenface: Eigenfaces are a set of eigenvectors used in the computer vision problem of human face recognition. The approach of using eigenfaces for recognition was developed by Matthew Turk and Alex Pentland beginning in 1987, and is considered the first facial recognition technology that worked. [5]

Eigenvector: Eigenvector of a given linear transformation is a vector which is multiplied by a constant called the **eigenvalue** during that transformation. The direction of the eigenvector is either unchanged by that transformation (for positive eigenvalues) or reversed (for negative eigenvalues). [4]

Eucledean distance: In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler. [6] The Euclidean distance between points $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$, in Euclidean *n*-space, is defined as:

$$\sqrt{}((p_1-q_1)^2 + (p_2-q_2)^2 + \cdots + (p_n-q_n)^2) = \sqrt{}\sum_{i=1}^{n}(p_i-q_i)^2$$

FERET database: FERET database is the de-facto standard in facial recognition system evaluation. The Face Recognition Technology (FERET) program is managed by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST). A database of facial imagery was collected between December 1993 and August 1996. In 2003 DARPA released a high-resolution, 24-bit color version of these images. The dataset tested includes 2,413 still facial images, representing 856 individuals. [11]

Facial Recognition System (fisherface): A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database**.** [12]

Hidden Markov model: A hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. A HMM can be considered as the simplest dynamic Bayesian network. [13]

Interest point detection: Interest point detection is a recent terminology in computer vision that refers to the detection of interest points for subsequent processing. An interest point is a point in the image which in general can be characterized where it has a clear, preferably mathematically well-founded, definition, a well-defined *position* in image space, the local image structure around the interest point is rich in terms of local *information contents*, such that the use of interest points simplify further processing in the vision system. It is *stable* under local and global perturbations in the image domain, including deformations as those arising from perspective transformations (sometimes reduced to affine transformations, scale changes, rotations and/or translations) as well as illumination/brightness variations, such that the interest points can be reliably computed with high degree of *reproducibility*. Optionally, the notion of interest point should include an attribute of *scale*, to make it possible to compute interest points from real-life images as well as under scale changes. [15]

Match point detection: Match point detection is the term used in computer vision that refers to the detection of the matching of the interest points between the given two images.

PCA: Principal components analysis (PCA) is a technique used to reduce multidimensional data sets to lower dimensions for analysis. Depending on the field of application, it is also named the discrete Karhunen-Loève transform, the Hotelling transform or proper orthogonal decomposition (POD). PCA is mostly used as a tool in exploratory data analysis and for making predictive models. PCA involves the calculation of the eigenvalue decomposition or Singular value decomposition of a data set, usually after mean centering the data for each attribute. The results of a PCA are usually discussed in terms of component scores and loadings. [20]

Three-dimensional face recognition: Three-dimensional face recognition (3D face recognition) is a modality of facial recognition methods in which the three-dimensional geometry of the human face is used. It has been shown that 3D face recognition methods can achieve significantly higher accuracy than their 2D counterparts, rivaling fingerprint recognition. [21]

1.4 References

[1] Grgic, Mislav, and Kresimir Delac. "General Info." Face Recognition Homepage. 2005. 21 Oct. 2007 <http://www.face-rec.org/general-info/>.

[2] Gross, Ralph. “Chapter 13. Face Databases” Handbook of Face Recognition Ed. Stan Z. Li, Ed. Anil K. Jain New York: Springer, 2005. 301-328

[3] Viola, P.; Jones, M., "Rapid Object Detection Using a Boosted Cascade of Simple Features", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Vol. 1, pp. 511-518, December 2001

[4] “Eigenvector.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation. <http://en.wikipedia.org/wiki/Eigen_vector>.

[5] “Eigenface.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation. <http://en.wikipedia.org/wiki/Eigenface>.

[6] “Euclidean Distance.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation. <http://en.wikipedia.org/wiki/Euclidean_distance>.

[7] "Face Detection and Recognition." Betaface. 17 July 2007. 20 Oct. 2007
<http://www.betaface.com/>.

[8] "Face Recognition Demo." My Heritage. 2006. 20 Oct. 2007
<http://www.myheritage.com/FP/Company/tryFaceRecognition.php?lang=TR>.

[9] "Face Recognition." Electronic Privacy Information Center. 5 Sept. 2007. EPIC.
20 Oct. 2007
<http://www.epic.org/privacy/facerecognition/>.

[10] "Facial Recognition System." Wikipedia. 18 Oct. 2007. Wikimedia Foundation.
19 Oct. 2007
<http://en.wikipedia.org/wiki/Facial_recognition_system>.

[11] “FERET Database.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation.
<http://en.wikipedia.org/wiki/FERET_database>.

[12] “Fisherface.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation.
<http://en.wikipedia.org/wiki/Fisherface>.

[13] “Hidden Markov Model.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation.
<http://en.wikipedia.org/wiki/Hidden_Markov_model>.

[14] "How Facial Recognition Systems Work." How Stuff Works. 2007. 19 Oct. 2007
<http://computer.howstuffworks.com/facial-recognition.htm>.

[15] “Interest Point Detection.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation.
<http://en.wikipedia.org/wiki/Interest_point_detection>.

[16] “Linear Discriminant Analysis.” Wikipedia. 18 Oct. 2007. Wikimedia Foundation.
19 Oct. 2007

< http://en.wikipedia.org/wiki/Linear_discriminant_analysis>

[17] Lowe, David. Distinctive Image Features From Scale-Invariant Keypoints. University of British Columbia. Vancouver: University of British Columbia, 2004. 2. Nov. 2007 <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>.

[18] Ozkan, Derya. A Graph Based Approach for Finding People in News. Bilkent University. 2007. 24-28. Nov.-Dec. 2007 <http://www.cs.bilkent.edu.tr/%7Eduygulu/Thesis/DeryaOzkanThesis.pdf>.

[19] Pissarenko, Dimitri. "Eigenface-Based Facial Recognition." (2003). Nov. 2007 <http://openbio.sourceforge.net/resources/eigenfaces/eigenfaces-html/facesOptions.html>.

[20] “Principal Components Analysis.” Wikipedia. 18 Oct. 2007. Wikimedia Foundation. 19 Oct. 2007 < http://en.wikipedia.org/wiki/Principal_components_analysis>.

[21] “Three-dimensional Face Recognition.” Wikipedia. 29 Dec. 2007. Wikimedia Foundation. <http://en.wikipedia.org/wiki/Three-dimensional_face_recognition>.

1.5 Overview

In this report, we will give details about the current system and the proposed system. In the current system part, we will give information about the process we have completed so far by explaining the algorithms we have tested and their results. In the proposed system part, we will explain the system architecture and the subsystems of the project. We will also mention the algorithms we will try in the future.

2 Current System

In order to apply methods properly that we investigate and analyze, firstly, we needed to have a face image database. So we used web crawling to collect appropriate face images. Therefore we have designed our own web crawler, which is focused on MyHeritage's actual celebrity database in use. The web crawler has visited MyHeritage's celebrity gallery periodically and has fetched all face images of one celebrity at a time. Furthermore, the web crawler is extendible to collect different type of images from various news providers. In database, all faces are labelled with names of corresponding celebrities in order to meet reliability requirements. During a few weeks, web crawler has downloaded entire database of MyHeritage with more than 31000 images of 3600 celebrities. All images have been downloaded our local drives and stored at RETINA project's server at the moment. All information related with images is stored in the database except original image files. Since MyHeritage's database only consists of face images, any face detection algorithm was not needed to be run. But these images are specifically optimized for web usage and they are formatted as GIFs. So they are compressed with a lossless data compression technique which can be converted back to original versions. Images collected from web have to be converted into portable pixmap (PPM) or portable graymap (PGM) because original images are optimized for web usage, which do not have required information for image processing.

In the next steps of the project, we plan to develop a new algorithm for finding similarities between faces. In order to see how well our algorithm will work we need a base line to compare the performance of algorithms. We decided to form this base line with existing algorithms and before implementation of our new algorithm we are testing existing algorithms to see their successes and weaknesses. The first algorithm that we worked on was PCA based Eigenface algorithm. We first tried to run this algorithm on the FERET database, since we had not yet normalized our own images. The details of PCA Eigenface Method are explained in the following sections. We obtained some basic results, but we faced some problems due to memory constraints, since the sizes of the pictures in the FERET database were too large. Therefore, we could not investigate this algorithm further for the moment and decided to explore Lowe's algorithm.

2.1 PCA Eigenface Method

This is a well-known algorithm for face recognition. This algorithm uses eigenfaces of normalized images to compare faces. However, the lighting conditions are really important for this algorithm. All the faces have to be in the same lighting conditions in order for this algorithm to work efficiently.

The characteristic features of the face which follow a similar pattern in every face are extracted by using PCA, and called eigenfaces. Any image can be represented by its corresponding eigenfaces. The eigenfaces are prepared by using the images in the training set. To compare the new coming images, these images are also first transformed into eigenfaces. Then the eigenfaces of the new image and the eigenfaces of the training set are compared by looking at the Euclidean distances between the weights of the eigenfaces. If the distance is less than a certain amount, then the image is determined not to be a face. Therefore, this PCA also does face detection as well as face recognition in this extent [19].

2.1.1 Calculating the eigenfaces for the training set

Firstly, all the images in the training set have to be in normalized form. The average matrix of these images is found and they are subtracted from the original images in the training set. The covariance matrix is produced by using these subtracted images. However, this covariance matrix yields too many eigenfaces (Principal Components) and eigenvalues. Since only a few of them are actually needed, a selection between the eigenfaces is done to reduce their number. After the covariance matrix is computed, similarities can be found between the unknown faces and the faces in the training set. The new image is transformed into its corresponding eigenfaces and the Euclidean distance between the weights of the eigenfaces of the new image and the images from the training set are calculated. As a result of this comparison, it can be decided if the given image is actually a face or not, by checking if the average of the Euclidean distance between the given image and the images on the training set exceed a value. Also, by using the Euclidean distances, the faces that the new image is most similar with can be found [19].

2.2 Lowe's SIFT Method

In order to perform feature extraction of the face images in the database, we use Lowe's SIFT operator which gives interest points of the given images. These interest points are used to perform matching between images. SIFT operator consists of 4 major stages which are Scale-space extrema detection, Keypoint localization, Orientation Assignment, and Keypoint descriptor [17].

First stage searches over all scales and image locations by using difference-of-Gaussian function. In the second stage, a detailed model is fit to determine location and scale at each candidate location. Keypoints are selected based on measures of their stability. In the third stage one or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. In the last stage of the SIFT technique the local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. [17]

SIFT technique compares a pair of faces with respect to their interest points and assumes the interest points having the least Euclidian distance as the correct matches. Although SIFT technique works well for finding interest points, it may produce false matches between pairs of face images. For example, SIFT method finds 90 and 85 interest points for the images in the Figure.1 and Figure.2. In the matching phase, SIFT finds 40 matching points as shown in Figure.3. However, some of these matching points are irrelevant and must be eliminated.

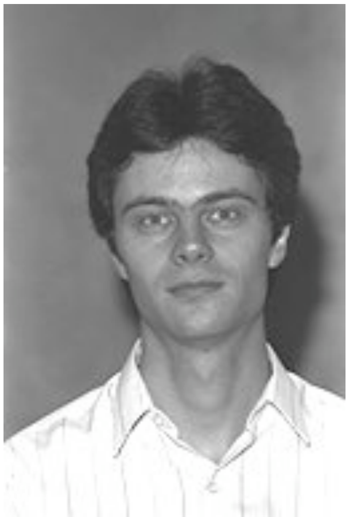


Figure.1 – Face1

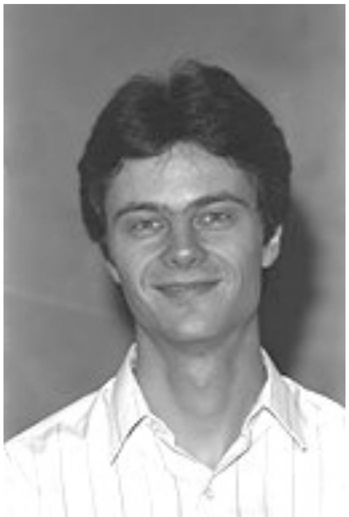


Figure.2 – Face 2

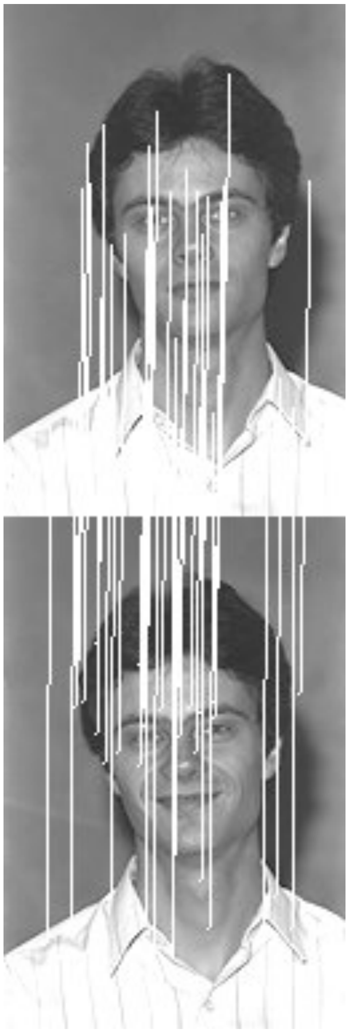


Figure.3 – Matching points

3 Proposed System

3.1 Overview

This project aims to find similar faces to a given face by using features of the faces ‘ the images. The system consists of two main parts. The first part of the system is responsible for creating the database to store the faces that are used to find similar faces and the features of these faces to perform comparison between faces. For this phase, the face photos are first converted into Gray Scale and then they are normalized for accurate comparison. From these normalized photos, the features are extracted and stored in the database. The purpose of this database is to extract the features of existing faces once. The system will deal with feature vectors of the photos instead of image files. By this way, querying time will decrease.

The second part of the system mainly involves querying and retrieval phase. The face in the given photo should be identified, normalized and be represented in Gray Scale for feature extraction. After the features of the photo are extracted, they are compared with all of the feature vectors that are stored in the database. The faces that have the most similar feature vectors with the given face are displayed as similar faces to a given face.

The system is designed as flexible as possible. There is room to make further improvements and changes on this system. The system consists of 3 layers which are user interface, application and database layers. Relations of these layers are shown in Figure 4.

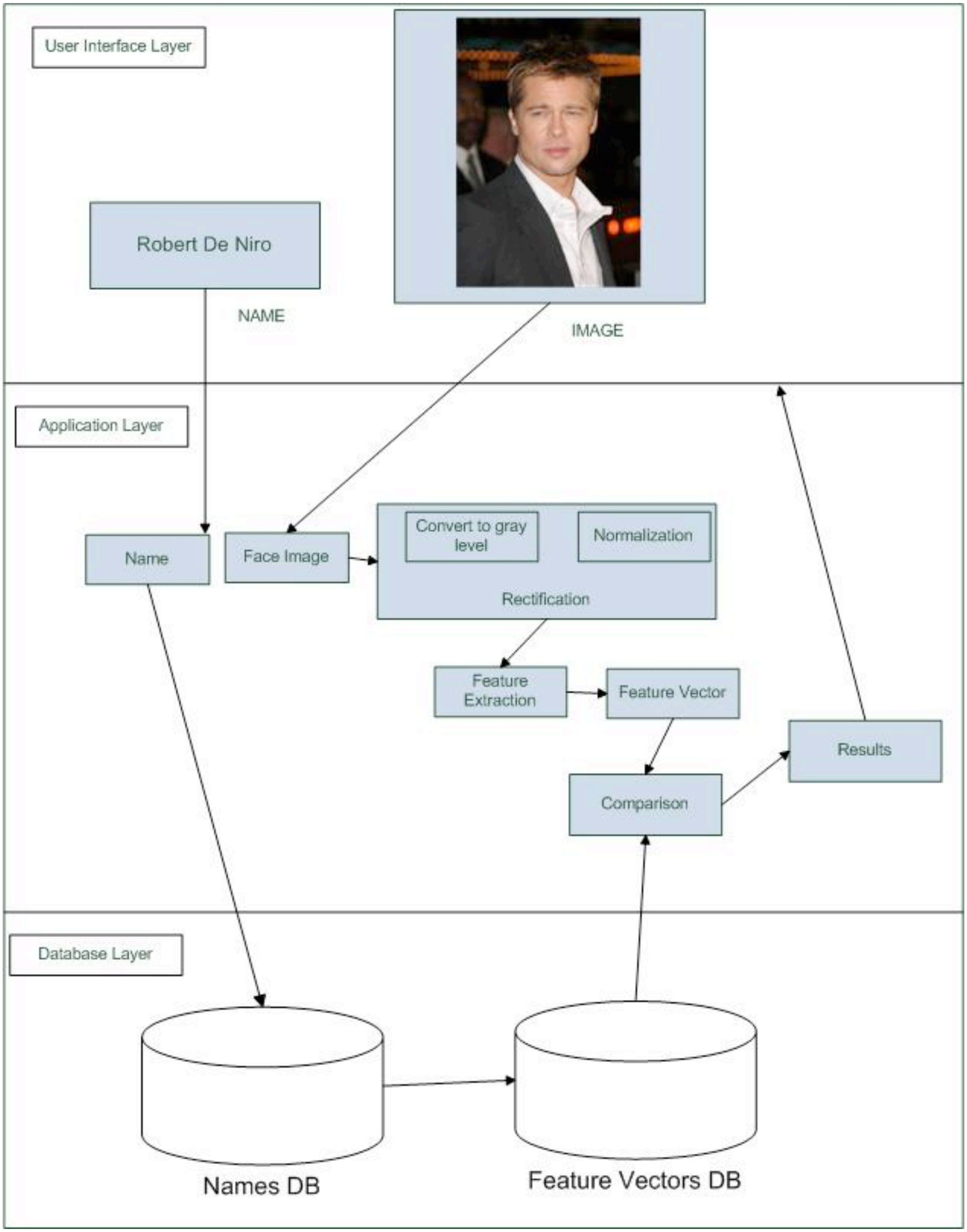


Figure 4: System Layers

3.2 Software Details

Relations of corresponding classes for application layer are shown in Figure 5.

Image: The class that represents the raw image given by the user.

Normalizer: This class normalizes the given image by scaling and converting the image color to gray level.

Face: This class represents the normalized imaged to be processed.

Feature Extractor: This class is used to extract the features of given image in order to perform matching operations.

Feature: This class holds the features of the images.

Matcher: This class compares the features of the images using the database.

Database Connection: It is used for the connection to the database from the application.

Query Result: This class holds the results of the matching between the given image and the images in database.

Celebrity: This class is used to hold the related information about the data in the database. In the user interface layer, the face images will be shown by using this class.

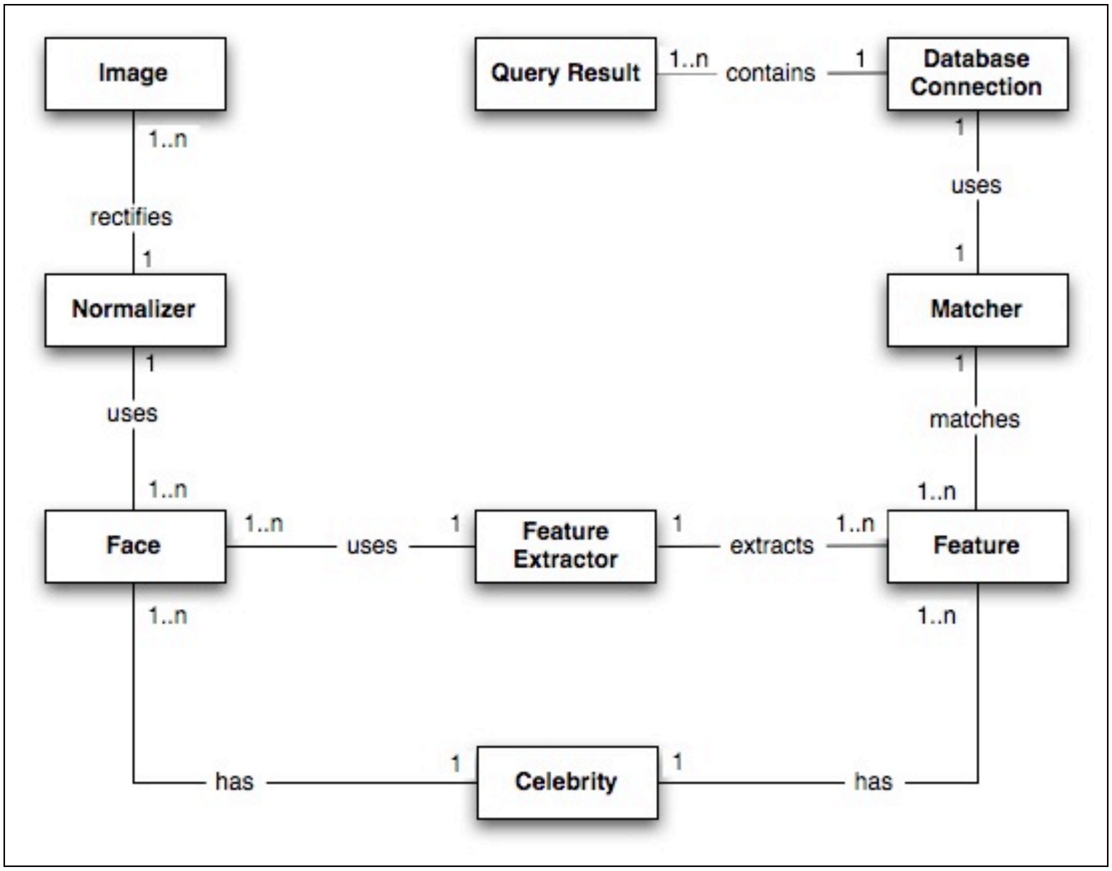
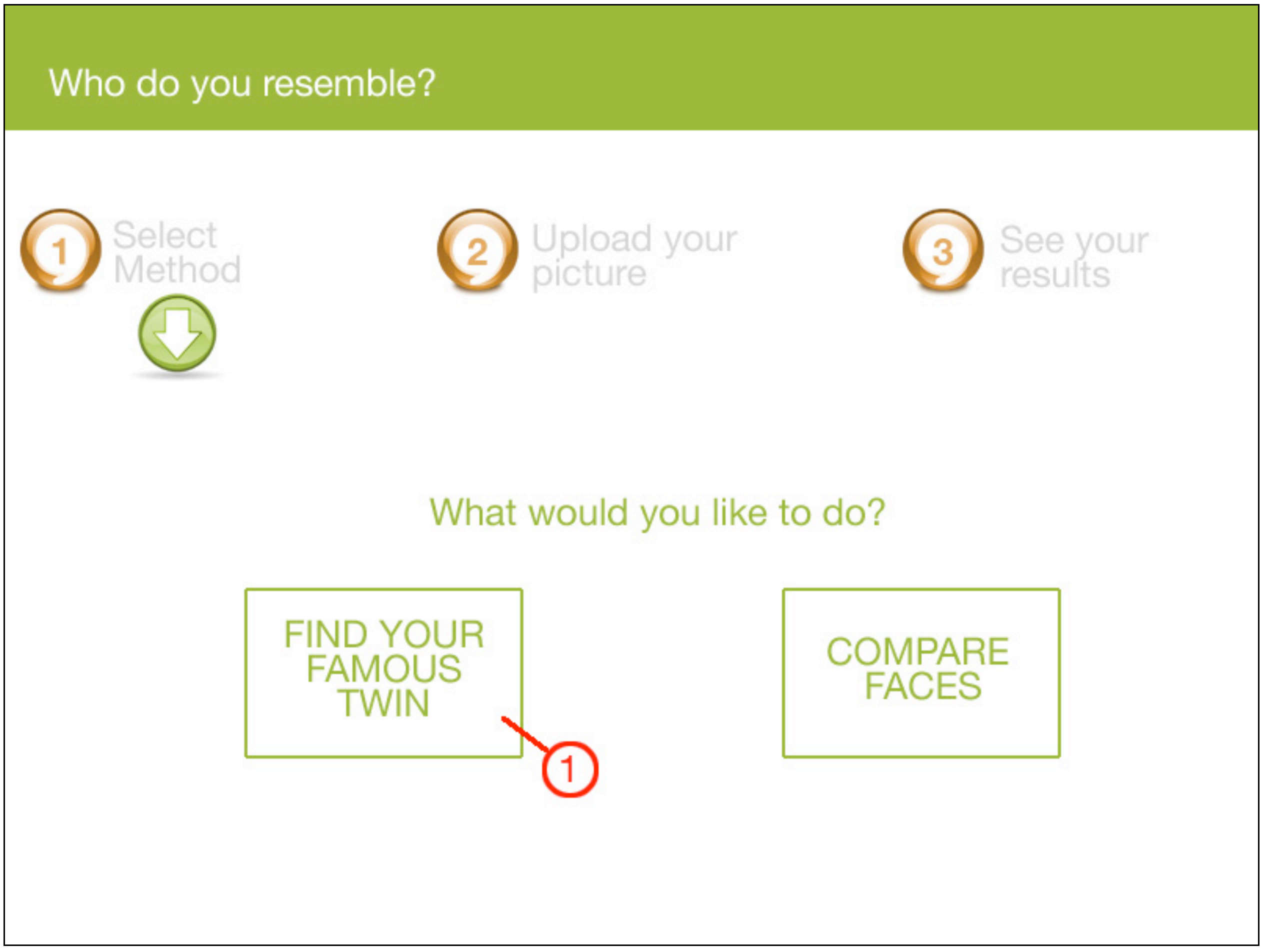


Figure 5: Classes of Application Layer

3.3 User Interface Design

The screenshots of the user interface for finding similar faces are given in Figure 6, 7 and 8.



Who do you resemble?

What would you like to do?

Figure 6 – User interface screenshots

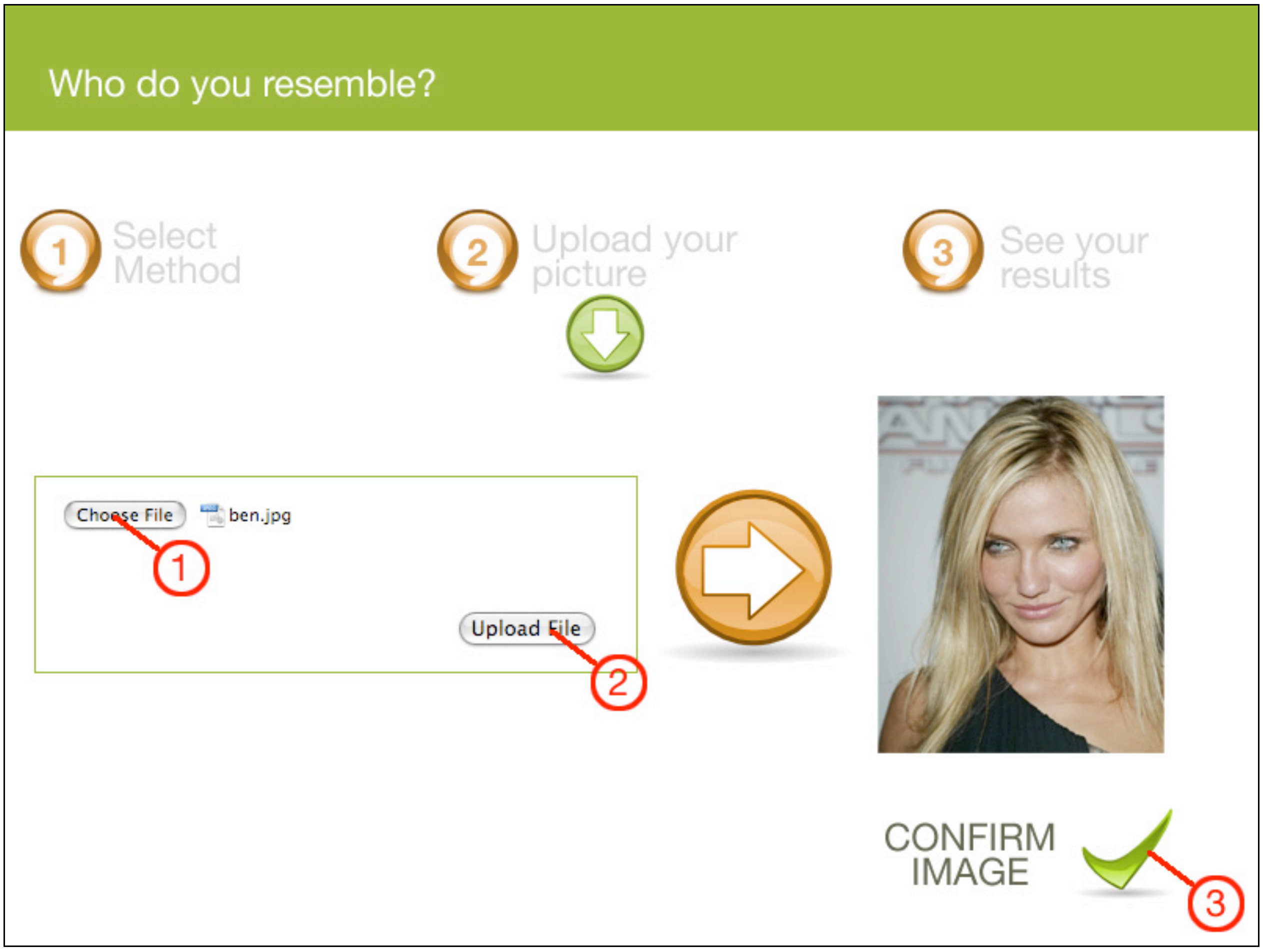


Figure 7 – User interface screenshots

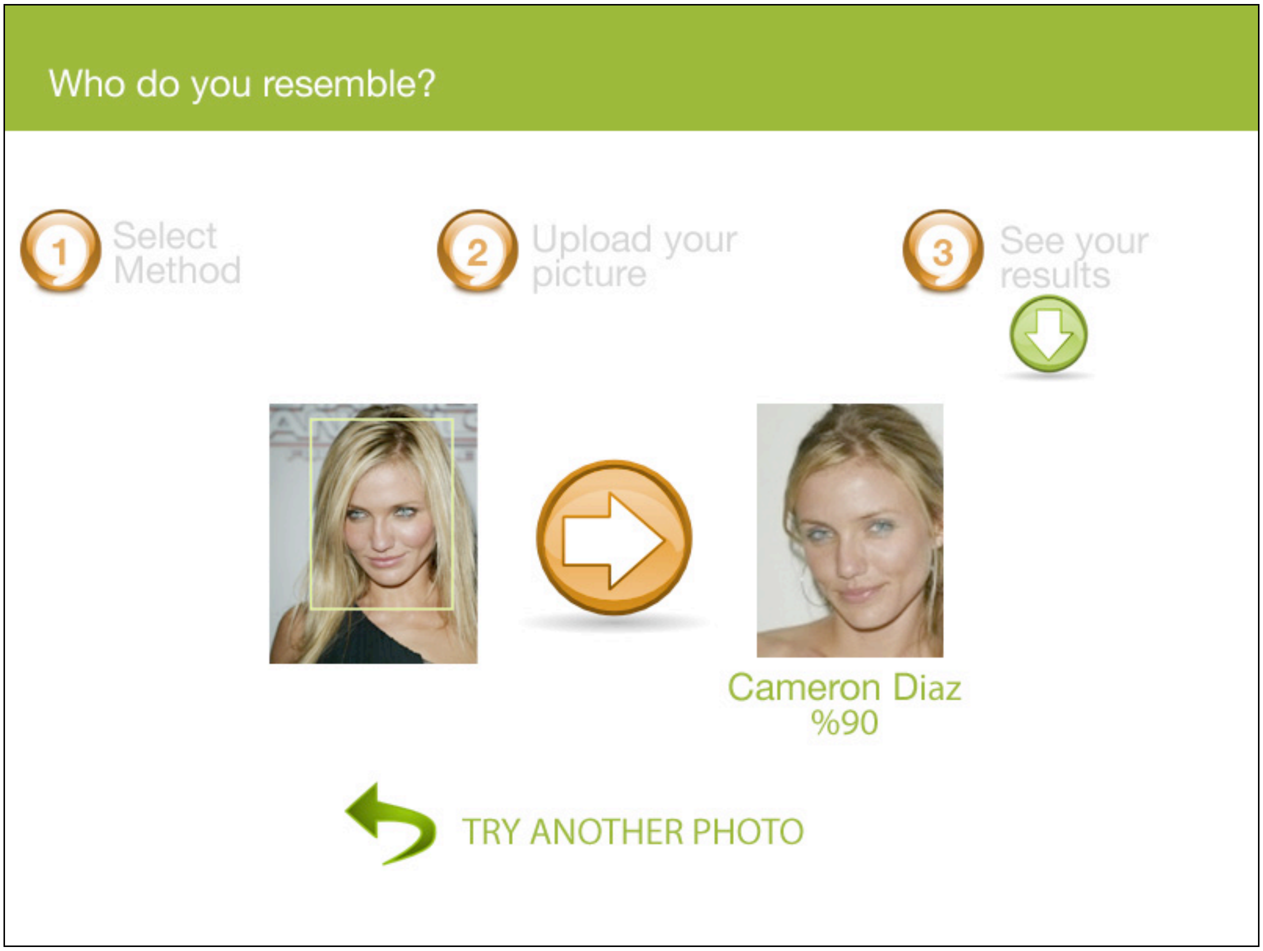


Figure 8 – User interface screenshots

3.4 Subsystem Decomposition

System decomposition is shown in Figure 9 as a deployment diagram and the following sections describe subcomponents of the system.

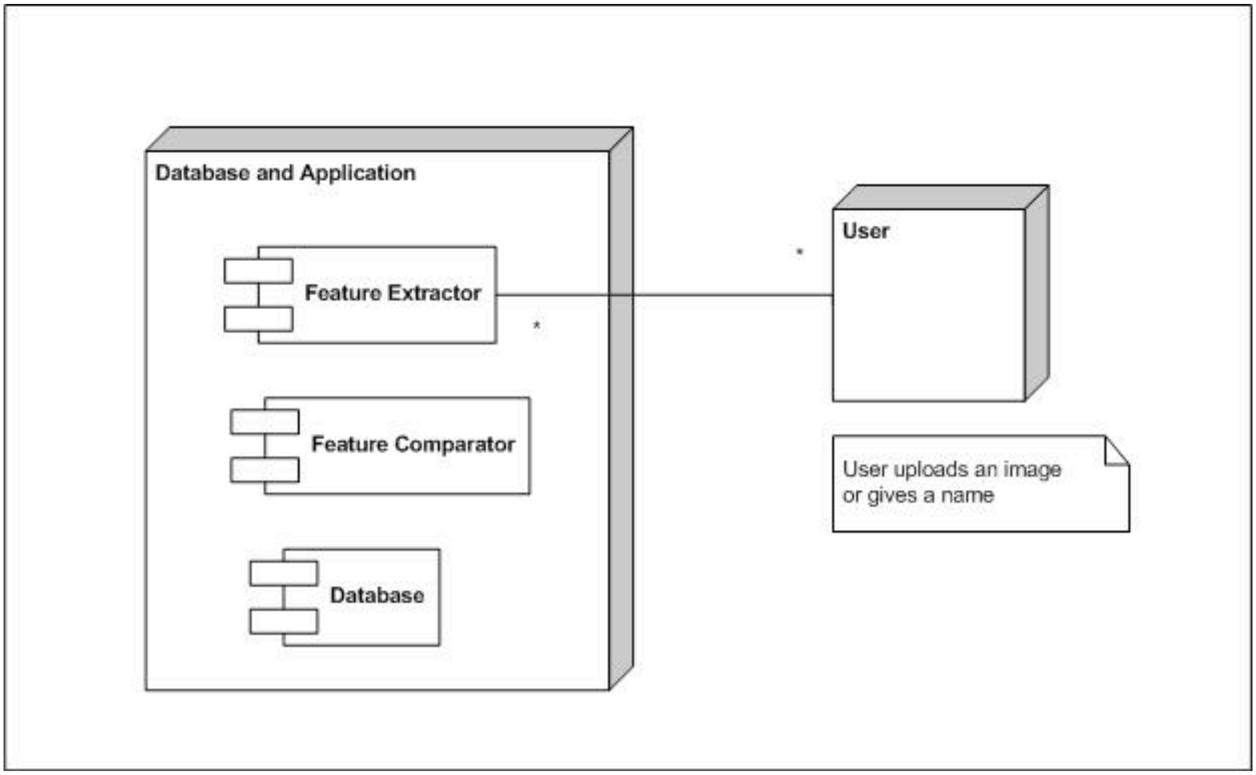


Figure 9: System Decomposition Deployment Diagram

3.4.1 Web Crawling and Normalization

3.4.1.1 Web Crawling

As described above in current system part, a web crawler has been designed which is focused on My Heritage's raw images of celebrities. A few photos of each celebrity are needed for a better and robust result set. The web crawler is designed for getting images from My Heritage's website at this stage; however, it will be a general-purpose crawler for future usage. It is planned to crawl also news with images and will get title, summary and image of the news from various news sources over Internet. Therefore, textual descriptions could be used while relating different faces among pictures. Web crawler is going to work during project development and database will be extended; therefore it gives a chance for maintaining an up-to-date database. During web crawling process, there will not be any face detection or elimination for raw images. So it will work the same way as general-purpose image crawlers. Additionally this web crawler will able to find the image by the given name if that person's image does not take place in our database. By this way usability of the system increases as well as it helps to enlarge the database with the required data by the user. More over, the web crawler will be able to add new images to the database depending on the search results of the users of the system. For example, web crawler can add more pictures to the database depending on the items that most frequently searched by the users. Finally web crawler will be used in order to update the database for some period of time.

3.4.1.2 Normalization

Performing normalization is required in order to increase efficiency of data sets and to eliminate unnecessary images in the specific item. This should be done in different manners, such as: During normalization, entire database is traversed. Image items, which have inappropriate sizes, are eliminated as a start. Later, faces are detected and rectified if necessary. The aim of this process is detecting several faces from a scene and making scaling process in order to make all images standardized, which is crucial for the rest of the process in order to get more accurate results. Therefore, every notable face will be cropped and standardized from raw image items. Another important point at this stage is converting RGB images into Grayscale, since colors are not important for face recognition at this project. Usually, skin color is not used as a feature. Therefore, color ranging between 0 and 255 is enough to extract many features from data sets. Eventually database consists of only faces, which are formerly normalized and converted into Grayscale. Since crawling more images might extend database, rectification process should be done continuously.

3.4.2 Feature Extraction and Database Construction

3.4.2.1 Feature Extraction

To implement feature extraction, we have already tested some existing algorithms as we explain in the current system. We will test well-known and well-recognized algorithms further. After comparing the results we obtain, we will find the best working algorithm(s), and improve them to get better results. The better and accurate the feature extraction phase is implemented, the better results we will get from the comparisons.

3.4.2.2 Database Construction

Database construction is an essential part, since querying faces will be done by using the database. We will make our database as flexible as possible and open for adjustments or additions for future improvements.

The faces will be kept in their normalized form; however, there will be a link to the original face image, before it was rectified, for every tuple. The feature vectors are stored for each face. Information about the person such as his name and age at the time the photo was taken is also stored.

Although, after finding similar faces for a given photo, the original picture and the name of that person will be revealed to the user, while comparing faces only the feature vectors will be used. We will also use this property of the database while testing the algorithms. Same person will have several different photos in the database. We will give a photo of a person who is already in the database and expect our query to return all the photos of that person on the database. So, our database will know that several faces of a person exist, but it does not use this information while finding similarities.

Another use of the database is that when a text-base querying is being done. If the user does not provide a picture but enter the name of a famous person, then the system will compare the other given face with all the images of the given famous person in the database and return the image(s) with the highest similarity rate.

3.4.3 Database Querying and Retrieval

This subsystem takes the uploaded image and normalizes it as explained in section 3.2.1. After normalization, feature extraction is applied to the image as performed during database construction. With the feature vector of the uploaded image, database querying starts in order to obtain the similar faces. In the database-querying step, the system compares the features of the given image and the images in the database. This comparison should be performed with an efficient algorithm that finds the best result. The algorithms giving irrelevant matches should be modified. For example, Lowe's SIFT algorithm gives false matches as we explain in the previous sections. In order to eliminate these false matches, Derya Özkan proposes two constraints which are geometrical and unique match constraints [18]. Geometrical constraint requires interest points to be at the similar positions when the images are normalized. Unique match constraint expects one to one correspondence between interest points of a pair of face images. This constraint removes one-way matches and eliminates multiple matches to a point. In the following paragraphs detailed information about these constraints will be given.

Geometrical Constraint:

This constraint assumes that the correct matches between images will appear at the similar positions. For example, right eye is expected to be around middle-right of a face in the normalized images.

In this constraint, a set of images is constructed which includes 5 images for each of 10 people. Then for each comparison correct and false matches are assigned in order to use them as training samples running on a quadratic Bayes normal classifier. Then these matching points will be classified as correct or false according to its geometrical distance. The geometrical distance corresponding to i^{th} assignment refers to $\sqrt{}X^2 + Y^2$ where

$$X = \frac{locX\,(i)}{sizeX\,(image1)} - \frac{locX\,(match\,(i))}{sizeX\,(image2)},$$

$$Y = \frac{locY\,(i)}{sizeY\,(image1)} - \frac{locY\,(match\,(i))}{sizeY\,(image2)},$$

and *locX* and *locY* hold *X* and *Y* coordinates of the feature points in the images, *sizeX* and *sizeY* hold *X* and *Y* sizes of the images and *match*(*i*) corresponds to the matched keypoint in the second image of the i^{th} feature point in the first image [18].

Unique Match Constraint:

After geometrical constraint is applied, some false matches may still exist. Generally, multiple matches of an interest point and one-way matches cause these false matches. Unique match constraint can eliminate these false matches by ensuring that if an interest point A has a matching with another interest point B in the other image then interest point B also has a matching with interest point A [18].

This method is an alternative for finding similarities between images. In the following steps of the project we will implement this algorithm to see its efficiency. We will try to decrease the database query and retrieval time as much as possible and give the best similarities.

3.5 Hardware/Software Mapping

The system will work over the Internet which enables users to upload their images and find similar images to these images online. Feature extraction of the given images and the querying-retrieval operations on the database should be as fast as possible in order to ensure high performance. Thus, the features of the computer used and the speed of the connection affect the performance of the software.

3.6 Persistent data management

We have collected our data by web crawling. Images in the database will consist of only face images. In the first phase, we took images from MyHeritage.com; therefore, we did not need to run face detection algorithms. However, after the testing phase and after finding reliable face detection algorithms, we are planning to broaden this data set, and get photos from news sites. We will identify and add these faces to the database, while keeping a link to the original images in the database. This is how the initial dataset will be constructed.

After that, while using the text-based querying if a user enters a name that does not exist in the database, then on the background the system will create a new web crawling task to extend the database to include face images of that person so that when that user comes back, the system would have added that name to the database and the query can be done.

3.7 Access control and security

Our database will be on the Bilkent Server, so the testing is done on the Bilkent Server where there can be no security violations. Only a few people are privileged to change the database since password is needed, so the database is safe.

Our program will be reachable by using the Internet, however, the users can only use the system, and they cannot change it. Therefore, our system is secure.

3.8 Boundary conditions

We will try to be as flexible as we can with the images that can be recognized and processed in the application. Surely, the most accurate results will be obtained from the faces that are in frontal view and in similar lighting conditions. We will try to allow as much perspective as we can, even accept faces from profile, however, there is a limit on the freedom of perspective for our application to work correctly. If the faces are turned around more than a certain degree, then our application may not give satisfactory results.

Also, it is really important for our project to work correctly when a face, which already exists in the database, is given. Since, we have several photos of the same person in the database, we expect our program to return these images as the most similar faces to the given face.