POLITECNICO DI MILANO

Polo Regionale di Lecco

Faculty of Engineering

Master of Science in

Mechanical Engineering

PERSONAL IDENTIFICATION THROUGH FACE RECOGNITION

Supervisor: Prof. Emanuele ZAPPA

Master of Science Thesis by:

Göksu USLU

Matr. 740325

Academic Year 2010-2011

Contents

1. Introduction

The face recognition systems have become a very popular personal identification system lately due to demands. It will be more successful and more popular in the future since it is fast, practical, and easy to use. Definitely the most important factor is the recognition success rate in face recognition and this factor will be better by the advancing technology in the near future. In order to imagine relation and thus the development between the today's and future face recognition systems, the background of this subject should be understood well. In this way we can predict something concerning the time interval between now and the future by simply looking back at the developments from the very beginning to until now. Thus it is easy to see that the personal identification through face recognition will be a very popular and valuable system in the future as well, and this emphasizes the importance of this subject.

In this work, personal identification by way of the face recognition has been studied. The goal of this work is to analyze different face recognition softwares in order to define the sensitivity of each of them to the image acquisition conditions and uncertainty sources including scaling, expression, viewpoint, and illumination. In order to separate the effect of each uncertainty source, the database of faces that has been used in this work has been created using software that is able to create 3D models and customize almost everything about the face; such as features, expressions and even make up. The relevant advantage of the use of database is that we can change each parameter separately; therefore we can analyze the effect of each of them.

In the next section, after the brief history of the subject, some fundamental knowledge has been given concerning the basics of face recognition. Because it was thought that it would be hard to understand some sections that would have been discrete without understanding how the system works. In the following parts, the rest of the work was studied from creating database to the results obtained by using this database in softwares.

2. State of Art

In this part the general information is given about the face recognition history and the fundamentals of face recognition. Evolutions in technology also reflected on face recognition and this made this area more effective but complex and due to this fact at least the fundamentals of the face recognition should be well understood in order to contemplate on results better. That is why it was thought it would be better to start from the history, application fields and to resume by analyzing the general approaches and methods superficially. Thus, this part is the basics of what operated in the subsequent tests mathematically by the softwares.

2.1 History of face recognition

The studies about face recognition started nearly in the middle of 60’s by the developing of computers. The many properties of our faces lead to use of different methods that can analyze these properties. The first method among those was the Eigenfaces method, which was suggested by Matthew Turk and Alex Pentland in 1987. In the past, the face recognition programs were just able to process by comparing any 2D image with another 2D image in a database. For those methods, one had to look towards the camera directly to increase the efficiency and accuracy. The variation in the light or not looking directly to the camera had caused to errors. Present-day, with the new technology, 3D softwares are being used.

2.2 Face Recognition System Applying Fields

A system which can recognize many faces can have many application areas such as in bank ATM security and police departments. For some applications face picture is the only data source for recognizing the people. Another example of the system is where the pictures obtained from the video records. In this kind of method the face might not even be seen clearly. Thus these systems can be used integrated with some technical methods for deciphering people’s voices and speech by computers. These sorts of hybrid methods have advantages in terms of reliability and time saving.

For the last two decades, face recognition is the subject that has been studied closely. These studies lead to success in face recognition and important advances have been done. Between appearance-based holistic approaches, Eigenfaces and Fisherfaces showed clearly that they are productive for the large databases. Regarding feature-based graph matching approaches are also successful, but, if it is compared to holistic approaches, it can be observed that feature-based methods are less sensitive in alterations in the lighting, observation point, expression changes, and to inaccuracies concerning localization of the face. These are the real advantages but the requirement of feature extraction techniques makes this approach likely to be erroneous in some parts [1].

In computers face recognition the mostly concentrated part is detecting individual features and clarifying a model by the size, position and relationships of these features between each other. In face recognition, a research showed that single features and their relations are inadequate to explain the performance of adult human face identification. However this way of face recognition (Carey & Diamond, 1977) is still the most popular one among the all approaches. In 1966, Bledsoe tried semi-automated face recognition with a system having parameters such as normalized distances and ratios between the points of eye corners, mouth corners or nose and chain character and the system was composed of a hybrid human-computer. Afterwards, a vector was developed having 21 features and faces, which used standard sample classification methods. Later, Fischer and Elschlager aimed to determine alike features automatically and used an algorithm for local feature template matching and a global measure to find and determine the features. Until the study of Yuille, Cohen, and Hallinan in 1989 this matching approach was resumed and made better. Their strategy was established upon deformable templates where parameter values are calculated mutual actions between the images. Finally we can talk about the three methods under the topic of appearance-based approaches. They are direct correlation, Eigenface Method and the Fisherface Method [2].

2.3 Eigenface Technique Basics

It is good to have some fundamental information on the subject of Eigenface for the reason that it is the core of our topic. Some formulas are also important and they are shown here for understanding the process from gathering the faces up to recognizing them.

Our initial step is to acquire a set S with M face images. If we want to use a training set of face images as; Γ1, Γ_2, Γ_3, ..., Γ_M. We can create our set as follows;

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \dots \dots, \Gamma_M\}$$

Then the next step is supposed to be obtaining the average face of the set, or in other words, "the mean image" that is defined by Ψ, in order to find the difference Φ between the input and mean images with the help of next formulas [8].

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n$$

$$\Phi_i = \Gamma_i - \Psi$$

Later we search for a set of M orthonormal vector that is defined by u_n. Here u_n, describes the distribution of the data best. The k^{th} vector, u_k, is chosen as;

$$\lambda_k = \frac{1}{M} \sum_{N=1}^{M} (u_k^T \Phi_n)^2$$

is a maximum, conditionally upon;

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1, & if\ l = k \\ 0, & otherwise \end{cases}$$

where λ_k and u_k are the eigenvalues and eigenvectors of the covariance matrix C which can be obtain such that;

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T$$

$$A = \{\Phi_1, \Phi_2, \Phi_3, \ldots.., \Phi_n\}$$

And it should be noted that A is an M by N matrix which are stating number of both pixels and images. Matrix C however is N^2 by N^2. For typical dimensions the task of determining eigenvalues or eigenvectors is unmanageable. For this task a more achievable process is applied [5].

When the dimension of the space is greater than the number of data points in the image, $M < N^2$, there will be $M - 1$ and not N^2, meaningful eigenvectors. The eigenvalues of the remaining eigenvectors will be zero. By solving a 16 × 16 matrix(by considering 16 face images) instead of the 16384 × 16384 matrix and later taking into account the linear combinations of the face images Φ_i, in the end it is possible to solve for the N^2 dimensional eigenvectors of an M by M matrix [5].

The M by M matrix is made after the analysis; $L = A^T A$, where $L_{mn} = \Phi_m^T \Phi_n$, and finally we get M eigenvectors v_l of L. The eigenfaces, u_l, are formed by linear combinations of the M training set face images, determined by these vectors.

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k$$

$$l = 1, \ldots \ldots, M$$

This analysis significantly reduces the required calculations, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In actuality, the training set of face images will be relatively small ($M \ll N^2$), and the calculations become totally controllable. The associated eigenvalues enable us to rank the eigenvectors according to their condition of being beneficial in characterizing the variation among the images [5].

2.4 Direct correlation method

The direct correlation method, which is also known as ‘template matching’, is done by direct comparison of the relative amount of the light emitted from pixels of the images of the face. The bitmap images of pixels are transformed into a vector of elements that defines a point inside a dimensional image space. The images that have common qualities are placed nearby each other in that space, but images having special distinction than the other are set some distance aside. For putting this concept to use we can determine the Euclidean distance among the image vectors to interrogating the indication of resemblance between two images of the same face or different faces [2].

2.5 Fisherface method

This system is called Fisherface since it is established on Fisher’s Linear Discriminant Analysis (LDA) that was founded by Robert Fisher in 1936. Cheng et al. come up with a method which utilized Fisher’s discriminator for face recognition and Baker and Nayar were the ones who developed a theory which was also built upon a two class linear discriminant. For Fisherface methods it can be said that it has less percentage of errors if we compare it with eigenfaces and it works better for the different lightning conditions. As another difference between fisherfaces and eigenfaces, we can say that Fisher’s LDA searches for the effective directions for discrimination, but Principal Component Analysis (PCA) searches for the effective directions for representation. Thus, we can say LDA is a statistical approach that is specific for classifying the unknown templates by working on the known templates. This technique aims to increase the contrariness between-class scatter but it aims to decrease the opposition within-class scatter [3] [4].

In this method, in order to understand how a facial image can change from one to another under the different illumination conditions, expressions and any small change of surrounding, we can start from calculating scatter matrices for both within-class and between-class and then total one in order to reach eigenvectors of the reduced scatter matrices and in the end we can end up by multiplying it with a projection matrix. We can use the matrix, found by multiplying those matrices, for the same intention where we use projection matrix in the methods that are based on eigenfaces [2].

2.6 Eigenfaces Method

Eigenfaces method was first used by Sirovich and Kirby to analyze the face more in detail and in a more effective manner. Sirovich and Kirby started from the face picture groups and determined principal components of these pictures. Later, they revived these images of the faces using the weight of the only small segments of the eigenvector (Kirby and Sirovich, 1990). They tested this method on database containing 115 face photos and they found that 40 vectors should have been enough to recreate a face with an approximately three percent error tolerance. After little while they developed their original method by taking into account the symmetry of the face (for instance eyes on each faces, nose, etc.) and they tested the algorithm on a database composed of 87 people. They used to Singular Value Decomposition (SVD) to achieve those results [10].

Turk and Pentland developed this idea further in 1991 and produced one of the first full automatic systems. They tried their system on a database which has the images of 2500 people. The images were selected from different dimensions and lightening conditions. Their system obtained right classification results for the 96 percent of the images, which was taken under the different lightening conditions. It also reached success for 64 percent of the images that were in different image scales.

In 1994, Pentland and Moghaddam also did some regulations such as using Eigenfaces Method with statistical methods in order to use a template for extracting the partial properties to recognize faces. After three years, Zang, Yan and Lades found that the Eigenfaces Method was getting worse dramatically when there was an extensive light alteration.

In one of the other studies which was done by O'Toole, Deffenbacher and Barlett in 1991, the relationship between the Eigenface and properties such as gender or race was examined. The race recognition is also important because of that many linear and non-linear templates were built. The templates have been used for both pixel-based and feature-based representations of the face as an input. Using the same network but with different sets, Fleming and Cotrell reached an accuracy rate of 67%, Golomb obtained a 91.9% concern with the accuracy of model generalization performance [9].

In particular, when it comes to face recognition, the Principal Component Analysis (PCA) is a commonly used method for reducing the dimension and by way of that we can convert each primary image of the training set into a corresponding Eigenface. This technique is about grouping faces that are distributed or finding the set of face images' eigenvectors of the covariance matrix considering an image as a vector in a dimensional space. Then we order the eigenvectors as each of them expressing another value of the alteration in the images. So we can think that the eigenvectors simply show the variation among the faces images. Hence, there will be an eigenvector affected by the location of images that we can show as a face so that we can call those set of vectors as eigenfaces. The eigenfaces are called as the best eigenfaces when they have the largest eigenvalues that represent the largest variance among the set of images. The most resembling images of the initial faces can be reached by using only those best Eigenfaces [2] [5].

In this method the faces can be reconstructed with the faces associated with known individuals that are expressed by Eigen picture weights. Thus, this sort of approach for face recognition includes three steps, which are: obtaining an initial set of face images (which is also called 'training set'), determining the eigenfaces from the training set and keeping the amount of images that have the highest eigenvalues and thus define the face space, and projecting each face images onto a face space to calculate the required dimensional weight spaces for each individual. As for recognizing the new face images, we can examine it in five steps, which are: projecting the input image onto each of the eigenfaces to determine the weights, controlling if the image is close enough to face space to decide whether the image is a face or not, sorting the weight pattern as known and unknown person if it is decided to be a face, and as the last two steps in an optional way updating the eigenfaces and weight patterns and finally calculating the distinguishing quality of the unknown face that is seen few times for uniting into known faces.

2.7 Calculating Eigenfaces

In order to calculate eigenfaces we can use set of M, N by N training images. Later we take into account each and every training image as a N^2 dimensional vector then find the average image. Afterwards, we find the difference between each image and average image and calculate the covariance matrix. We resume by calculating its eigenvectors (eigenfaces) and eigenvalues. To achieve this calculation we normalize the eigenvectors and we choose the eigenvectors that have the highest eigenvalues. If the eigenvalues are high, we can say the more typical features describe the particular eigenvector. Finally we can end up by neglecting eigenfaces with low eigenvalues since they describe just a small section of typical features. After the highest valued eigenfaces, eigenfaces can be determined and this is the end of the training phase of the algorithm [5].

3. The Virtual Database and Facegen Software

This section is about how the database was created by using the software, and the software's itself. In order to define the sensitivity of the softwares, the different kind of new faces should have been created as being a new database. The Facegen software was used for that, and, at first it is better to examine how the database was created, and, moreover, this part was thought to be a need to strengthen the aims we try to reach by analyzing the functions of the Facegen software.

3.1 Facegen Modeller Software

For each program of face recognition there is one database needed. Most of the programs are with their databases and give good results, but only using a totally new database can eliminate the doubts concerned with performances with different databases. Thus, a new database was supposed to be created and this is achieved by software for modeling of the face of a person. This software is known as Facegen Modeller.

Facegen has been developed to make it possible for a user to create faces quickly. By using this software it is pretty easy to create a face in a short time. If random faces are needed with a single button named "generate" under the "Generate" menu. Under the "Generate" section there are some other options which exist for generating the faces from different races. There is an also "All Races" button for creating faces among the different races and gives the results randomly in terms of different races when it is clicked. After obtaining those faces, it is feasible to change all other characteristics, or by using the "Set Average" button the faces changes can be changed all over again and the initial face can be created when the program was first run. Thus it can be said that "Generate" button is the main part of the section named "Step 1".

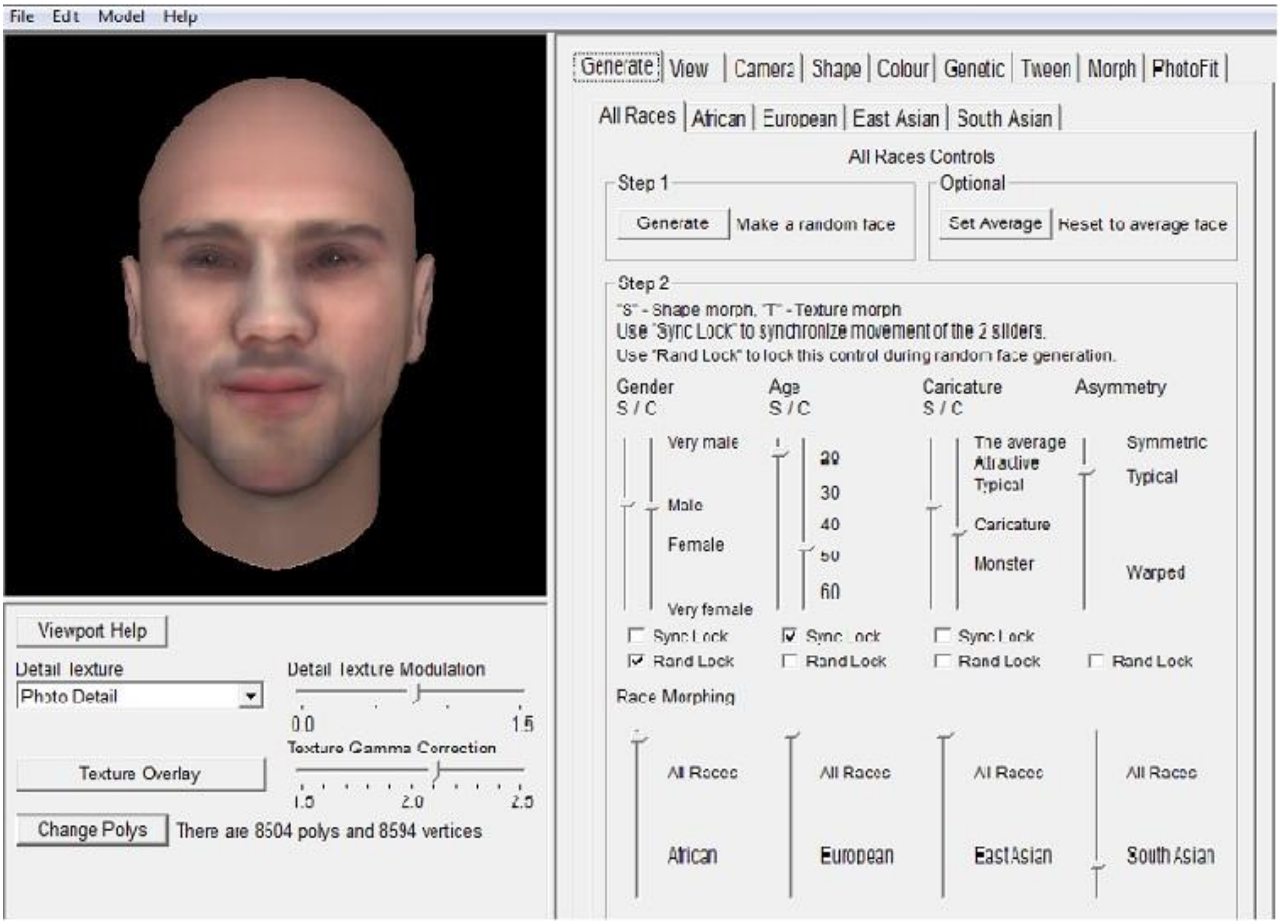


Figure 3-1. Main Menu of Facegen Software

With Facegen software, it is possible to change the age of the faces as well. For instance, if it is wanted to get a human face belonging to a 50-year-old woman, the result can be reached easily. This is simply done by using the sliders within the section called "Step 2". It shall be noted that there are two sliders for each property such as gender or age. These sliders are named as S and C and represent "Shape" and "Texture" morphs. The meanings of those are simply as follows: the shape morph is for the shape or the geometry of the face and the texture morph is for the texture or the facial skin of the human wanted to be created. For instance, when we want to create a face of the old man, we can drag the sliders to the up on the male or up to very male for getting a proper male face first. Then, to change the age of this face we can use the sliders for the age. But here the importance of those sliders can be understood well.

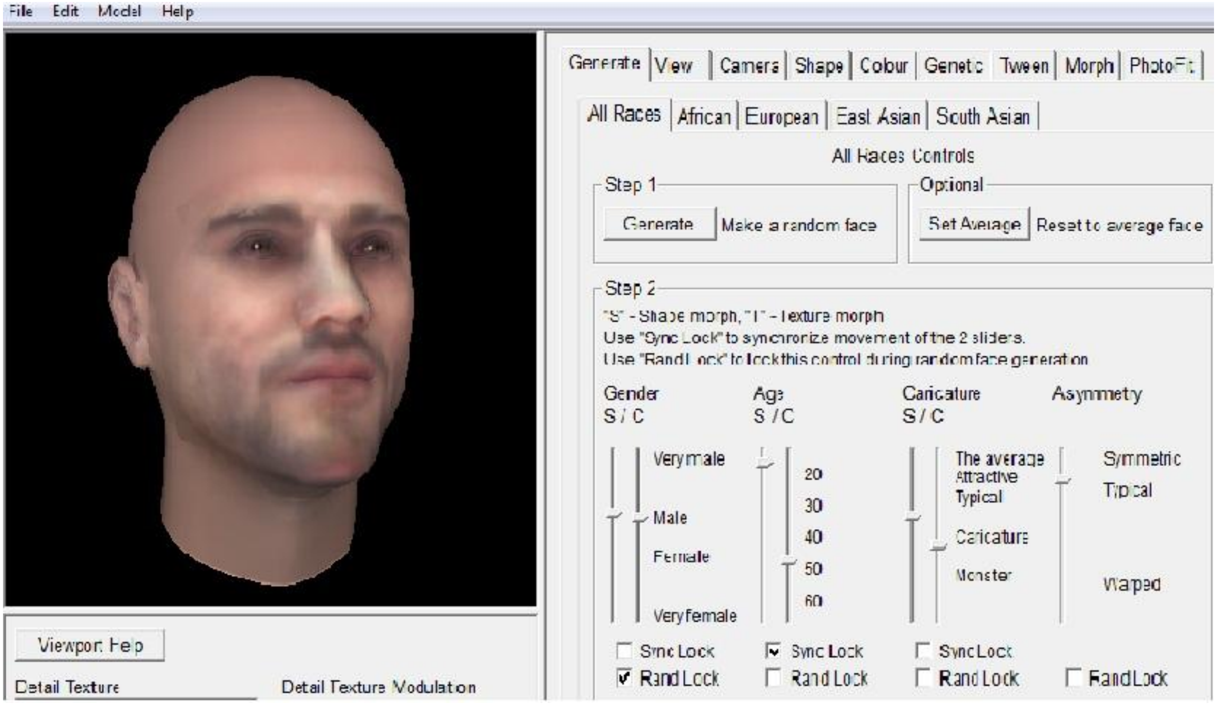


Figure 3-2. Generate Section of Facegen Software

The slider on the left is for the shape, which represents the shape of the head or skull for the gender case, but for this case it represents the alteration on the features. The alterations are very interesting such as the change in the nose, ears, lips, chain and even in the eyebrows. These changes for the nose and ears are about the size so when we push the sliders down the nose and ears are getting bigger also cheeks fail to keep their fullness since they are belonged to an old face. These alterations are clearly and can be easily done by Facegen software so it gives very realistic results. Because also in reality it is common that the some features of the head get bigger since they are actually cartilage. As to other slider, it is simply for the texture of the face, as already can be understood from its name, and as we drop down this slider the color of the face changes and even develops an older look. Moreover, deep wrinkles appear around the mouth, nose, and also the baggy skin under the eyes (in another word, 'pouches') become clear just like they would do during real aging processes. This function is also available for symmetry, gender and realism (caricature).

In sections such as shape, color, morph there are other sliders for even successively nostrils, make up, and expressions. It is also possible to create a shorter face with heavier eye brows, and to change the looking direction of the pupils. It is even possible to change the expression of the lips as though created face pronounces “eh” sounds or other sounds. This feature adds a quality to the expression or completes the expression of fear or astonishment. Another good part of this software is that expressions can be made asymmetrical. For instance, when we want to put any winks to the eyes we can do it separately from one another, blinking only one of the eyes as the other one remains on its initial position. So the result we can understand is this program offers great variety of different functions for creating realistic faces.

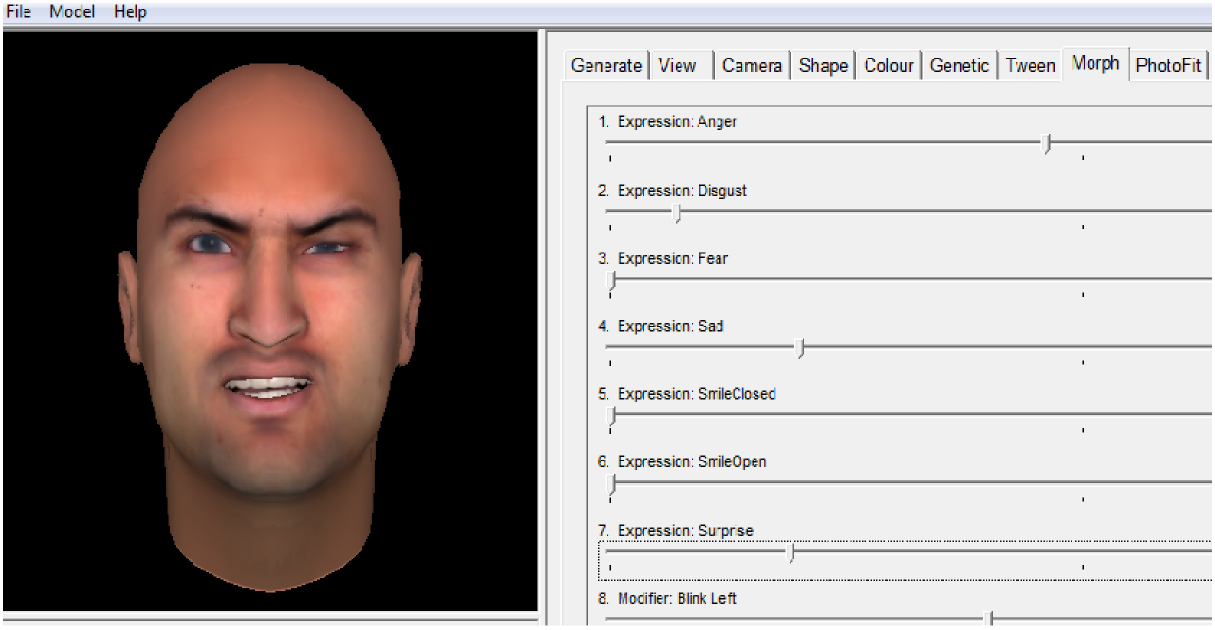


Figure 3-3. Morph Section of Facegen Software

As for another aspect of this software, it presents the possibility of creating faces from rare physicals, for instance the heads of famous people. So, with this system it is easy to get celebrities’ faces and to change their features as needed. It is also possible to use someone’s face directly for using in the database. This system is well prepared for the users of Facegen and is named Photofit. With this tab, it is possible use exact real photos. In order to do this, first of all, the photo should be taken clearly and the mouth should be closed. Moreover there should be no expression on the face, and the eyes should be open. Glasses or scarf (if it covers the some parts of the face) must not be used; hair must not be hiding the some feature of the face. Other details that are not going to make difference or warnings about the other subjects should be avoided. However, the Photofit tab works anyway but these are the warnings in order to get maximum efficiency from this tab. This option is taking time though and the requirement of three images (one frontal and two from the sides) makes this system hard for creating huge databases. For instance, if it is the face of a famous person, it is almost impossible to find the images from the proper angles or form the sides. Also the pitch angle gets important, thus, for these kinds of faces, it is nearly impossible to obtain the desired conditions for the images that are going to be used for the Photofit option and because of that. After putting the images we are supposed to assign feature points to define specific points, such as the corners of chin, center of the pupils of the eyes and so forth.

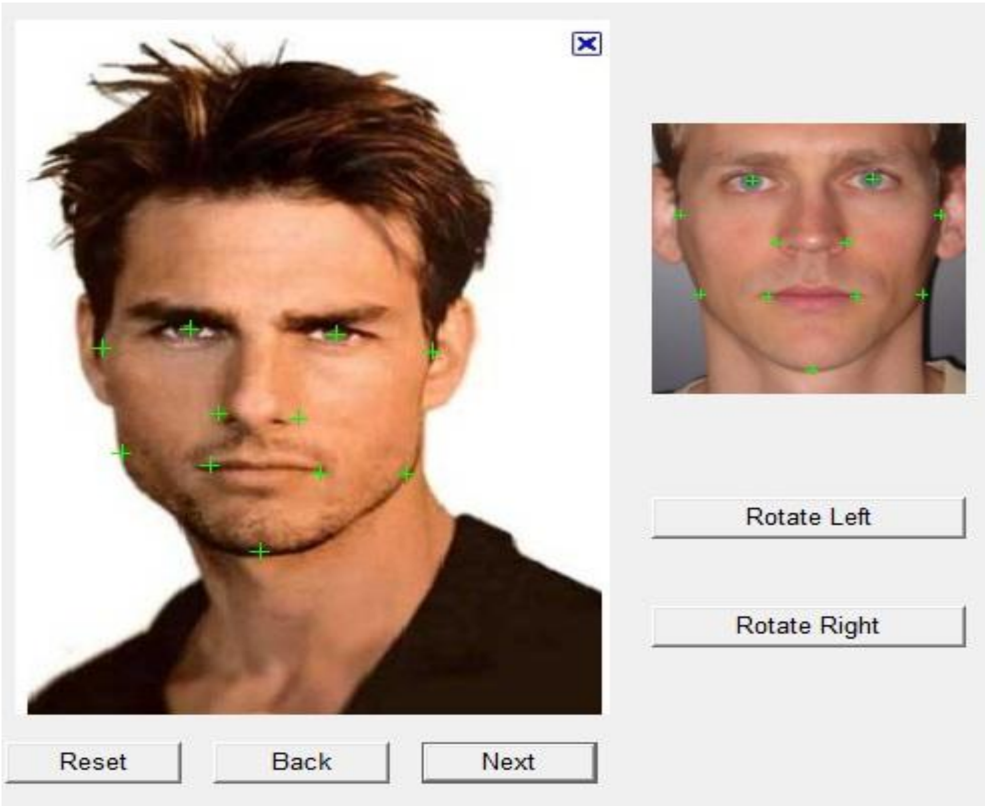


Figure 3-4. A PhotoFit Application by Facegen Software

The actions are also supposed to be done for the other images from the sides. Afterwards, it is written in the program that it would take two to five minutes in respect to the performance of the computer. Actually, in respect to the skin texture of the image, it may take more than that when we check the “preserve facial hair in detail texture” box, which means the software will also take into account the skin surface of the faces. In the end, with this section it is possible to use images to create the face and they can be saved as 'facegen file' and, whenever needed, the face can be tried under different conditions, such as different illumination, pose and expressions. There are also other options to improve this face by later operations which are extra charged.

3.2 Creation of the New Database by Facegen Modeller Software

By using the Facegen Modeller software a database was created for our use because of the requirement for those images of faces. For using the images over and over again, when needed, we better try to get the source first. From this source we can get the images anytime. For instance, for some of the source codes the initial images can work flawlessly, but this does not prove that the program is 100 percent successful. Because of this necessity the database had to be completed.

In order to create this source database we tried to do as much as different type of faces due to probability of errors might be gotten easily. This is achieved by keeping us away from using the "Generate" button in "Step 1", which makes random faces. Instead, the use of other option (in "Step 1" again), the optional "Set Average" button could have been used after completing creations of each faces, because this button enables us to start our work over again by making a new face from the very beginning. If the other button is used, and the different faces were created with that way, there would be a chance of coincidence of the similar looking faces, so it was better to spend more time on faces and get realistic faces and then results.

We began by "Set Average" button to start making a new face all the time. After that, by gender sliders first we can specify the sexuality of the humans we were trying to make their faces. In this part there are two sliders; 'shape' and 'texture' as explained before. Under those sliders there are two boxes named "Sync Lock" and "Rand Lock". As it can be understood from the names pretty easily they are the boxes for locking the sliders. Upon the sliders it is already stated their function since Facegen Generator is a user-friendly computer software. Thus “Sync Lock” is for synchronization of the motion of two sliders which means moving them together equally. As to “Rand Lock” it is for locking the both sliders and preventing their movement while we use the random face button (the Generate button). This means that if we want to create a random face, but meanwhile if we want to use same age or/and gender, these boxes help us to change other things about the face while the properties of interest remain same.

After the explanation about functions of the boxes we can make some comments about also the sliders. The first one of the sliders for the gender is for the shape of the head which means both shape of the skull and shape of the features. After our decision was made, it was easy to see the difference between the ends of the sliders. For the shape slider, whenever we move the slider up we see that the face is getting more masculine shape as the skull contour becoming sharper. On the contrary, when we move the slider to the bottom we see that the face counter gets gentle and the keen edge disappears. With regards to other “texture slider” taking place for the change of skin, as we move it up the facial hairs around the jaw appear and the eyebrows become thicker. However, when we do the opposite action the software reveals them back and the skin color get lighter. So these functions were the first ones used for our database.

The second pair of sliders was used to define the ages of the faces. Thus, the faces have different ages and some of them even have different shape and texture ages. Therefore, this function can be utilized to get realistic results if needed, since, in reality due to natural aging processes and other affects, people might not have skin that belongs to that specific age. For the third pair, we see caricature sliders. These are about the realism of the faces and when we move one of them down the face gets far away from a real face and develops a shape like in the cartoons. Next to those pair of sliders there is an “Asymmetry” slider. In reality the human face is generally not symmetric and this slider has a limit named typical which corresponds to more realistic faces, so this function was also utilized while the database was being created. Below these function there are race morphs which were used to decide about the races of the faces created. In order to get realistic results these four different sliders work in an interconnected way, thus, when we move the first one the others also move automatically. It can be noted that it is also possible to choose one of the races easily from the other tabs next to the “All Races” tab. By the help of these tabs the generation of the face could have been concluded roughly and for more details the “Shape” and “Color” buttons could have been used for our case to complete the most important part of our database. In both tabs we can apply both symmetric and asymmetric changes to the faces. For the “Shape” section we can make changes on the shapes of features, cheekbones, chin, brows, and temples. On the other side, for the color button we can change the color of those, moreover eyelids and even eye whites. Finally from the left bottom of the page we can change the detail textures when we want. This feature was also used for our faces in order to create faces which are close to real human faces. Because by using this feature we can add real texture from the samples which contain some wrinkles, even scars and birthmarks. These are not the only changes we can make of course, but these are enough to make a comment about the general properties of the Facegen Software. Thus these functions are used for our database that was to be created. In fact, after following these steps mentioned above for each face and after small changes when needed, the faces were saved one by one after finishing each one of the faces and to start the other one the “Set Average” button was used and finally the first part of the database was finished. At the end of this part there were forty-four faces saved. The rest of the images for database were all created from these images.

The next step would be the “View” section, which is for adjusting the lighting source positions and ambient brightness. With this tab it is also possible to add a background photo and adjust its transparency. With the “Camera” section we are able to change the camera options, for example FOV (Field of View (or also Field of Vision)) angle, distance ratio, yaw angle, pitch angle. Among those, the yaw angle is very significant since it is playing a very important role in face recognition. The program starts with a yaw angle of 20.05 degree, which is not suitable for us for the first try. Thus, for the first frontal images of our database this slider was shifted to 0 degree.

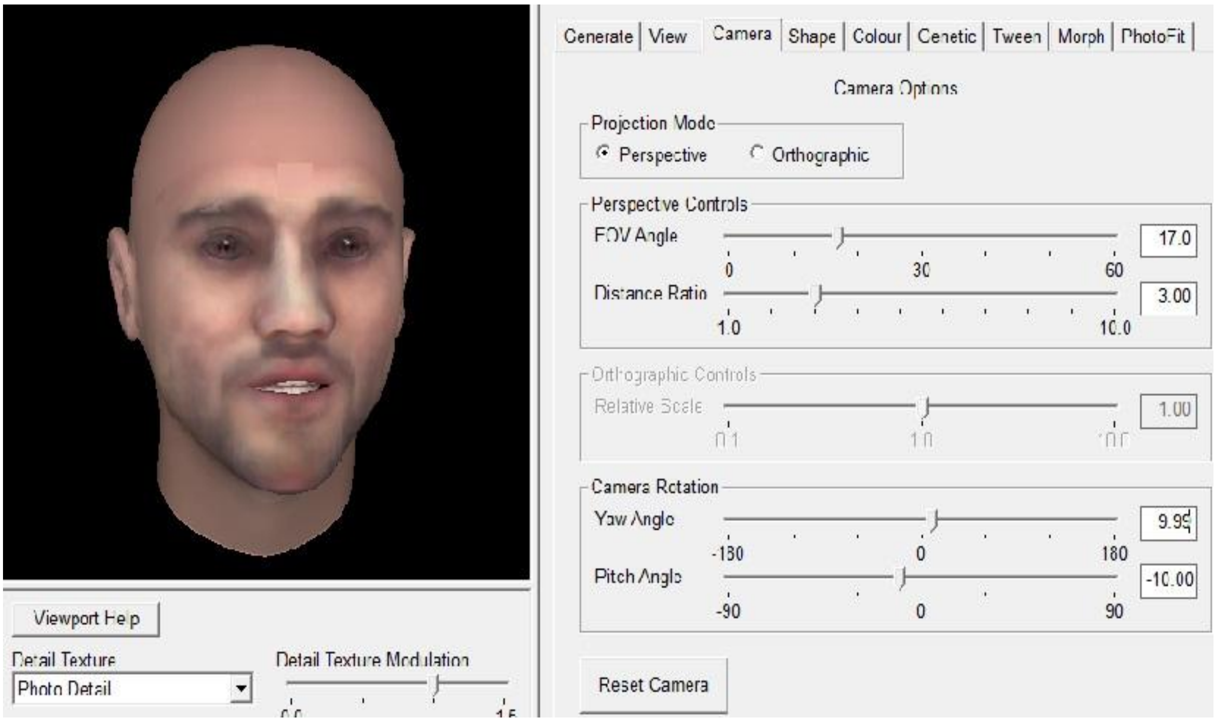


Figure 3-5. Camera Option of the Facegen Software

When we would like to create random faces that are similar to one face, we can use the "Genetic" tab of the software. The randomness of that similarity is dependent on "Randomness" slider that has a value interval from zero to one. After creating some face it is therefore possible to set randomness and click the "Generate" button. Then Facegen generates eight more faces which are very similar to the original one and we can choose one of them and keep on occupying with only that face. "Tween" section has nothing to do with the "Genetic" section since we do not simply get an image respect to what we put (by load target button) but we get a mixture of the face at the left, that is already generated manually by us, and the target face. Hence, we get a new face, which is very similar to the target image and the other image of course, and we are able to see it from three various points of view. If we desire to increase similarities between them we can move the sliders properly and we can get the results at the left part as our new face for later use. But this option seems to give unrealistic results when a real human face is entered to the software by the Photofit option, because when we open this sort of face and want to get this image's similar one of another female face and then when we move the slider down we see that at the left part there is a woman face with beard and also with the scars of the initial face. This case was tried by the

34th face (file name 34c.fg) of our database and even the scars of this face passed to its similar model since the software sensed those texture differences as innate characteristics of the face. Hence, it is not a useful tool when we think about to do this operation between opposite genders.

One of the most important sections of this software is the “Morph” part as stated before. This section was used during the production of our database as well. The c indices (e.x.1c, 2c...) represent those expression changes in our database while the numbers indicate the number of each person in the database. For each person the expressions are made differently and detailed. Furthermore, for most of the faces more than four or five sliders have been used to create more realistic face expressions. And finally, suitable phoneme sliders were used to improve the final expressions.

Thus, if we want to summarize how the database was created, we can say that it was begun from creating faces and setting the yaw angle to zero and finally saving all images from the faces with no indices. Then these conditions were set and later, by reducing the ambient light, the “a” group is completed and these images were saved with indices. Subsequently, the ambient light was set to its original and the “Yaw Angle” was set to thirty degrees to the right for each of them, then the pictures were saved gain and this group was called as “b” so that the indices were defined as “b”. The next step was the setting the angles to zero all over again and giving expressions to the faces which would have had the indices of “c”. Later it was thought that the angle of thirty degrees might be a little bit higher and the tests might be somehow inadequate thus more images were added such like under the change of five degrees of “Yaw Angle” and thirty (but minus since face turns towards down) degrees of “Pitch Angle” and these properties were named as “f” and “p” consequently. Thus our training database was completed with 40 different faces, and due to many different conditions, it reached many more images. As for the test database, some other different faces were also used.

4. Performances of the Face Recognition Softwares

In this chapter five different types of face recognition approaches were tested. This section is the most important part of our work and it forms the results that we will analyze in the following parts in order to make a comment upon the request of our goal. In later parts the results will be collected under the two main sections, which are 'ideal' and 'realistic' in order to understand how the softwares are being affected by different conditions and to comment on softwares' performance.

4.1 Programmes Tested in Ideal Conditions

In this chapter each of these softwares will be briefly introduced and tested in ideal conditions. Here the "ideal" conditions means images acquired from a front view and without any scale change. The same database is going to be used for also the other part which is called realistic conditions in order to get a better comparison.

4.1.1. Drexel University Face Recognition Program

The main targets of the program are to detect a real face and to recognize a human face as the initial purpose. General sections of the program can be roughly classified as using the set of images, which depends the result of interest, then inputting a known image which is already in the set and observing the Euclidean distance which will be compared with other results afterwards when we input an unknown image [6].

In case of recognition of a real face, we start using a set of images of faces. Subsequently, we input an image that is already among the images in the proper set. As we repeat the process by other images in the set, we start to get two results for each process as graphs that indicate weights and Euclidean distances of the input image. Moreover, we are able to get accurate results as of maximum and minimum values of the Euclidian distance graph. By collecting these results, we can get an idea about the images in our set, and then use them for comparison when we use images that are not present in the set, i.e. an unknown image.

For this program it was useful to show the results for both with initial database and the new database. The reason for that was to understand the real aim of the program and not to justify it according to the results we would have had in case of the use of our new database.

4.1.1.1. Drexel University Face Recognition Program with New Database

In this part the program was tested with the new database. In fact the result is, when this program was tried to find a human face in ideal conditions, the system does not work well as can be seen from the test results. These results are shown on the table where “S” indicates the same image from the initial database and “un” indicates unknown human face from out of the initial database which is used for finding the eigenfaces. Under this title the frontal images are tested. The test database is composed of two main parts of known faces and unknown faces. Since the program gives numerical results that are required to be compared later, the same images from the database are used initially.

The program works by taking the initial faces and calculating eigenfaces of them. Thus the forty different faces were used to reach this aim for our case. Then the first six images were used again in order to understand the results and to evaluate the other results, which will be achieved when we use images out of the database. As it can be understood from this aim the four different people’s faces were used afterwards. Among the results the first three tests give hope to us that we can find the faces easily because the small difference between maximum and minimum value which we might not suppose as a valid case for our last four tests since they are different images from database.

When it comes to next after the fifth test, for the last one for among the database images, we understand that difference between max and min value does not have to be small for even same images were being used. Just in case of seventh test we understand this program is not good for understanding human faces for even ideal conditions and should give more irrelevant results for real conditions. This is because of the difference between max and min value which is not more than 3000 which was more then 5000 for the previous test. The following results for other unknown people's images are the same in this way it is confirmed that the good results cannot be obtained by using this program for recognizing different human faces.

Test Image	Max Value	Min Value
S1	51936	50102
S2	52566	50291
S3	51628	50282
S4	52328	49910
S5	52730	50343
S6	55960	49442
un1	52698	49538
un2	51417	49547
un3	51610	50100
un4	51335	49624

Chart 4-1. Results of Drexel Software for an Ideal Case

As a result of this part, it possible to say that this program, showing the results detailed and properly, does not give good results for the case we test the real human images in ideal conditions. However, it is good to detect not faces and animal faces as can be understood from the following test.

4.1.1.2. Drexel University Face Recognition Program with a Present Database

In this part the tests were made for a different purpose, which was not personal identification, but it was useful for us to understand the aim of the program well and to get some other observations as well. Thus, system's original database was modified and some other images were added including artificial (such as a human wearing a mask) faces, an animal face, and some image that does not belong to a human being.

TEST	FACE	MAX VALUE	MIN VALUE
1	For a human face within the set	14934	12142
2	For a human face within the set	14633	12098
3	For a human face within the set	14434	12071
4	For a human face within the set	14213	12053
5	For a human face within the set	14626	12076
6	For a human face within the set	15883	12080
7	For a human face within the set	14768	11956
8	For a human face within the set	14992	11965
9	For a human face but not in the set	15178	13552
10	For an animal face	16540	15647
11	For an artificial face	15643	14979
12	For an artificial face	16574	15947
13	Not for a face	15031	14283

Chart 4-2. Results of Drexel Software with the Present Database

As for the tests, for the ones within the set, the program gives results at a vicinity of 11600 minimum value and this value is very close to others'. The maximum value does not have exact values as said in the tutorial of the program for which is about 15000 max value for the real faces cause it sometimes barely exceeds (just for one case it reached to 15883) that so-called limit and in order to understand whether an image is a real face or not it is better to evaluate both maximum and minimum values. For the images, which do not belong to a human face, max value rarely takes place around 15000 limit and sometimes hits the values greater than 16500 while min value is never below 13500. Furthermore, the differences between min and max values are always very close to each other, hence, best evaluation can be made by analyzing the both two values for real face recognition. There are some results for both real human faces and others on the table. It can be observed that for any image that is not present in the set the reconstructed image will be distorted. Also min value is considerably higher than the assumed one because it was around 11000 or 12000 for the faces inside the set. For the ninth test, we see that the minimum value is even higher than 13000 so from now on we can understand if the minimum value is too higher than 12000 this means that the face we want to test is not from the training set, which is called ‘unknown’. Since the minimum value is greater in these cases, the difference between max and min values are going to be smaller compared the ones from the training set of course.

For artificial human faces resembling human faces or for faces that we can call the products of imagination the system also works well to distinguish them from human faces and those faces result in more similar values. As a result it possible to say that this program, showing the results detailed and properly, gives cautious results and it is well designed to apply the Eigenface approach to recognize someone’s face.

It can be noted that when we analyze an animal face we see that the min value is much higher but as to comparison with an image which are not faces, the animal faces have more similar values to humans’ and this is caused by they have particular parts of the face as human faces do. On the tenth line which states for the test of a face which is belonged to an ape it can be observed that the minimum value is much higher compared to all human faces even though it is not in the training set. The value founded is 15647 and this limit is so far off from 14000. Therefore, from now on it can be understood that if for a test image the minimum value is much higher than 14000 it can be called as not a human face. This program is quite successful for the detection of human faces and even for the artificial faces, which are similar or human-like faces, because even for artificial faces that look human the results are greater than 14000, and nearly around 15000 as can be directly seen in eleventh and twelfth tests. On the thirteenth test there was a fighter type aircraft image tested. The values are for this test are very interesting because min values are much closer to ones in the test of humans’ images, but from the values it is easy to understand this image cannot be a face. The values of images might have been gotten like these because of little similarity to human face of the aircraft trunk from the profile. In the end, the program is works quite well for its purpose [8].

4.1.2. BPN (Back-propagation neural network) based Face Recognition Program

This method normally gives similar results as PCA based softwares because in BPN method the PCA is used as dimensionality reduction tool and by using PCA, the input data is decreased to a lower dimension to ease the classification and then this process is followed by a classification stage which is done by applying the vectors to BPN classifiers to obtain recognized picture [11]. Thus, this should be why it takes much more time by matlab. Efficiency of the Face Recognition System by using Back-propagation Algorithm is high so this algorithm is further than GA (generic algorithm) [7]. The system originally uses the bmp file extension type which keeps quality, but, since it is not compressed, takes up larger space compared to Jpeg & PNG. However this software seems really complicated because of its codes. When we first run the program system starts doing iteration.

In case of ideal conditions, the program reflects well to the user and actually gives good results. In fact, except for one mistake, the system found all the others right and found the right people's image among the all images within the training database in our tests.

Figure 4-1. An Incorrect Result from BPN Software

However, if this program is compared to others it can be said that the operations take more time, but it still gives really good results.

4.1.3. Eigenface Face Recognition System by Amir Hossein Omidvarnia

In this program there is an original database made up of colorful images. Among the original test images there are some images that were also used as training images. Hence, when an image is tried as an input image the system may give the correct result, but this is because using the same ones which were used for training as we can consider as an ideal condition.

However, even in these conditions the system can make serious mistakes. Sometimes the results are totally irrelevant than the other input images and the program simply cannot separate the genders. In the original software there are twenty images in the training database to be tested and when we increase the amount of images the results are not given properly. This condition will be explained in later sections.

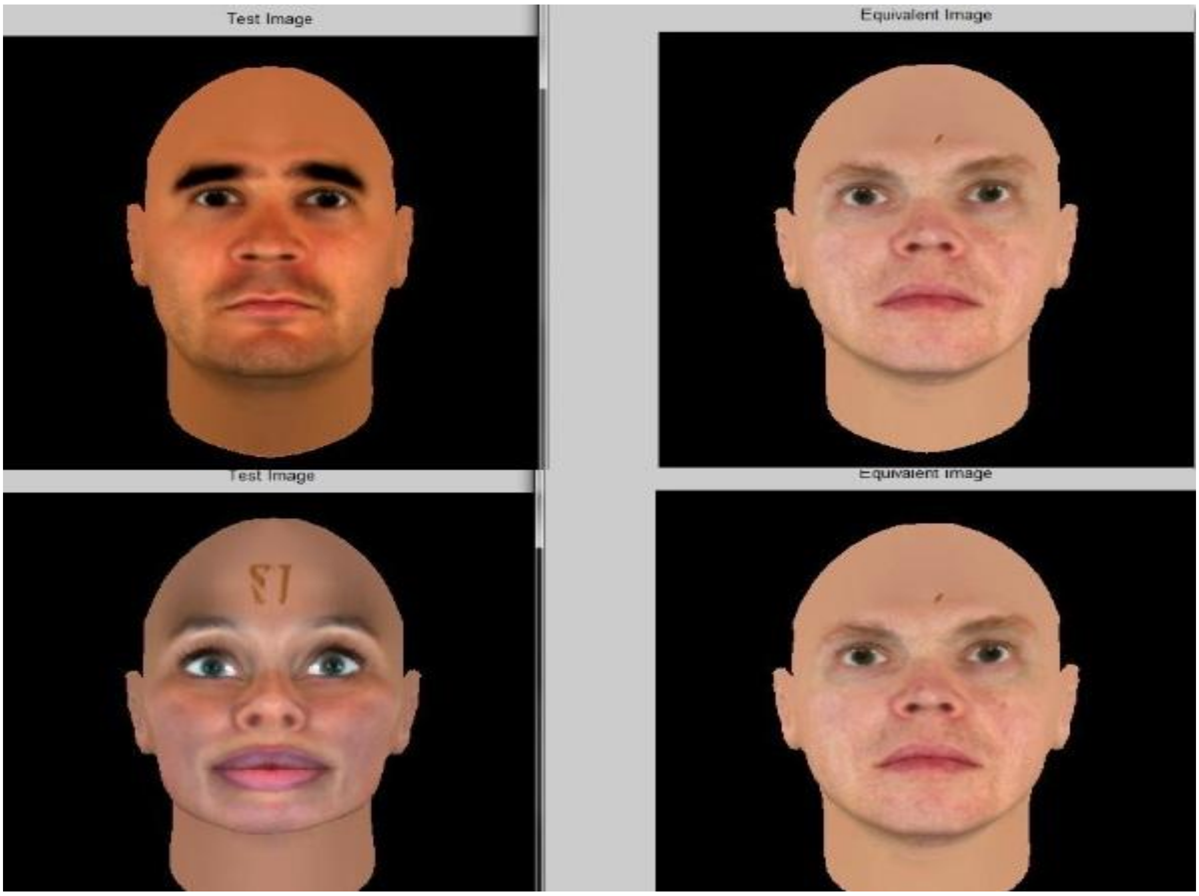


Figure 4-2. Some Incorrect Results from Eigenface System of Amir Hossein Omidvarnia

As it can be seen from the image, the program gets stuck with some images from the training database and whatever the input is, for the results the output gets become two or three images from the test database. Among these conditions it is not strange for us to see correct results because we obtain the equivalent image such as seen on the right of the picture and this means when we use this image as an input the program will give the correct result but this result also same result gotten by inputting the other images so we cannot count this as a real "correct result". Finally as the first point of view the program is unsuccessful to recognize human faces.

4.1.4. Fisherface Face Recognition System by Amir Hossein Omidvarnia

This system carries out a renowned FLD-based (Fisher Linear Discriminant) face recognition technique called Fisherface. The program is modification of the previous one to this method and gives much better results compared to that one. The system composed of each training and test databases as the previous one. The system does not require gray scale images.

Fisher algorithm is a refinement of the Eigenface algorithm. The program first compares two faces and also calculates the Euclidean distances between them then later find the corresponding image by searching for the one which has minimum distance among them.

The results of this system are far better in ideal conditions when it is compared the other systems. Among our database the system does not make any mistakes and is able to catch right image each time. Thus for realistic conditions, in regard to others, much better results can be obtained with this program. However, it must be noted that there are twenty images in the training database while we perform the tests and this is in order to have a better comparison in the beginning. But our real training database, created by forty images and meaning the double, was used for the some other softwares. In order to have better results for comparison, we used the first twenty images of the training database and this case increases the success rate of these programs created by A. H. Omidvarnia a bit. But this case will be compensated in the end and a well performed fair crosscheck will be done.

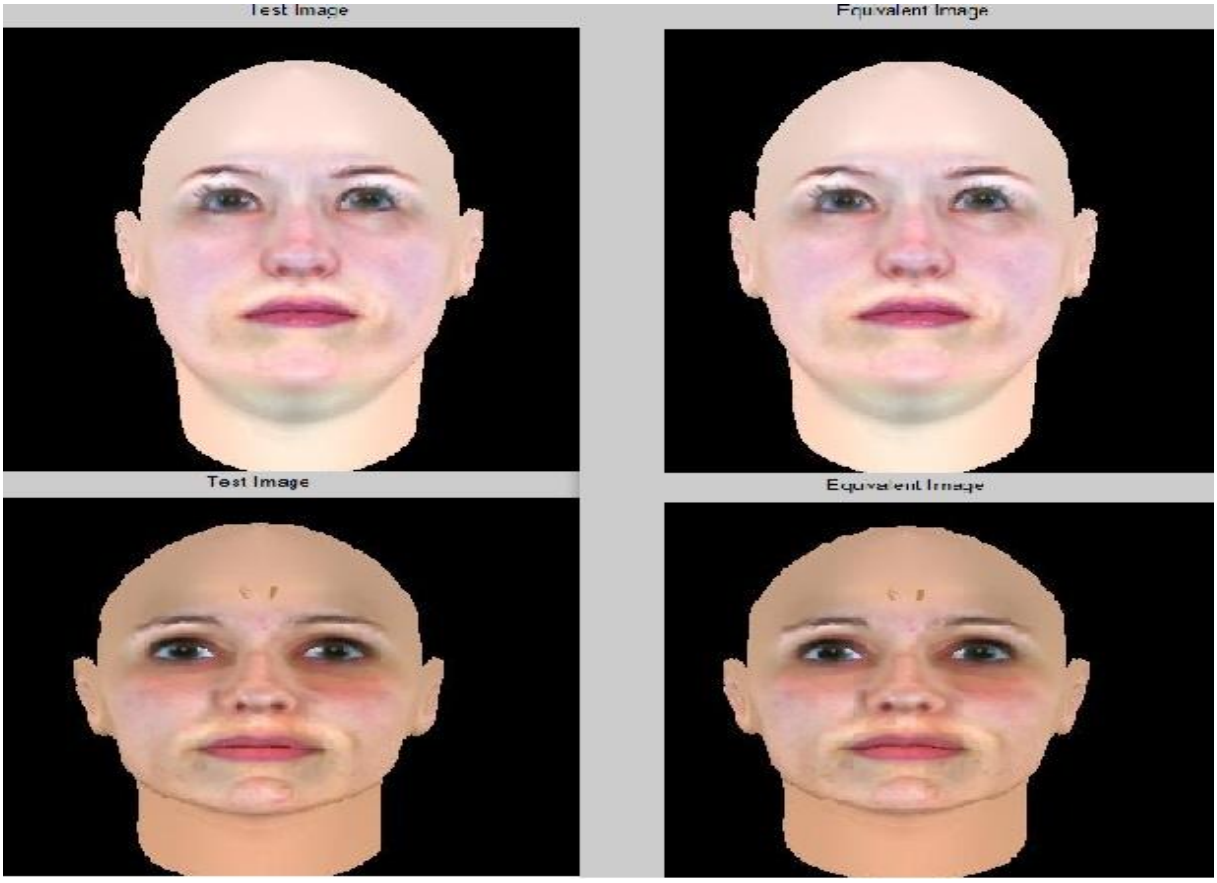


Figure 4-3. Some Correct Results from Fisherface System of Amir Hossein Omidvarnia

4.1.5. Face Recognition System by Zubair Saifi and Suvidha Sadhu

This is software which uses Eigenface approach for face recognition. When we open the software we are met with a nice menu. On the main menu we are able to choose different operations. For a general database first we press "1" and the program asks us how many images we would like to continue with and after this option the system starts to calculate eigenfaces. Then we start to wait for this because it takes a little time. With this way, in other words, we load the database and from that point on for this program there is no need for loading database again if we do not press the wrong button. The next step to do is choosing the input image from database and for that we are supposed to press "2". Then all of a sudden a new window appears where we can choose the test image and then it gives the result quickly. To repeat the operation we press any key and then press "2" again. There are also options for updating and deleting the database or to quit the program moreover from matlab. But since these operations can be done also manually, there is no advantage of using those options.

Figure 4-4. A Correct Result from the System of Zubair Saifi and Suvidha Sadhu

Because of the menu and the first part of the program, which is about finding the eigenvalues, the expectations get really high for this program and it lends wings to the user. This promising software tested for our database first, but for the ideal conditions, and the results were exactly as expected and the system gave the right results all the time. There is an also another function of the software. It is possible to put name for every person whose images in the database and when we input any of the image of one person and if the system is able to find which person that is (it finds perfectly in ideal conditions), it will give also the name of the person and also possible to show other information like age of the person. Thus if this system can be developed in the future, this idea is going to be the one used for the security of for many other aspects.

4.2 Programmes Tested in Realistic Conditions

Within this section the same tests will be repeated but this time the conditions will be different. The conditions are called realistic conditions and the effect of realistic conditions will be more important for us due to hardness of providing ideal conditions in real life. The “real” conditions include one or more of the uncertainties in the image generation here, such as different angle of view (i.e. not exactly any frontal image), scaling of the image size (i.e. simulating the changing in camera-face distance during acquisition), and expression.

4.2.1. Drexel University Face Recognition Program

This software was tested by a large database again with the generated faces. The first part of the database was composed of different faces. Each of them had 3 more different forms under different variables such as illumination, expression on the faces, and the angle of the camera. The second part of the database was for the test including randomly selected poses of the same faces. Moreover, the faces there are not in the database and also the same images that were used in the first part of the database. For this program, first part of the database is for obtaining the mean image. Before that the training set is converted into normalized training set so the all images are normalized in order to lessen the error occurred because of the lighting conditions and different backgrounds. After obtaining the mean image, program compares it with selected input image and reconstructs the image and shows both of them on the screen. In another screen there are two graphs that show the weight of input face and the Euclidian distance of input image. Then we also get the minimum and max values. These three results enable us to comment and decide about the result. Here are the values that were gotten after the test,

TEST	FACE	MAX VALUE	MIN VALUE
1	6.bmp	56845	54600
2	7.bmp	57901	54799
3	10.bmp	57244	54219
4	21.bmp	56947	53535

5	34.bmp	56441	53204
6	38.bmp	55936	52944
7	39.bmp	57128	54089
8	40.bmp	56207	53959
9	40b.bmp	57389	54356
10	40c.bmp	56005	52925
11	unknown1	54619	51851
12	unknown2	55035	52316
13	unknown3	55972	52631
14	1same	51936	50102
15	34same	52550	49832

Chart 4-3. The Results of the Software from Drexel University for the Realistic Conditions

The tests made from "1" to "10" state for the different poses of the human faces that created by facegen. From "11" to "13" the unknown faces tried which have no relation between the faces in the training database and finally "14" and "15" are the same images that were used in the training database. As it can be observed from the values, that were found for the faces which are in the database, the max value does not go under 56000 (it just fell right under this limit value for once) and for this case the min value is high. Between 11 and 12 tests the max values are too low. When it comes to thirteenth one there is no clear difference between the results for this test and the results from the tests at which the ones done with the images that are present in training database. So this one can be counted as a mistake or we can say that system is not able to recognize the face and does not give an appropriate result in this way. When the same images are used the system gives lower values that are really close each other so for exact images which are used in the initial database, the system works perfectly. However results found for the last two tests also do not give us any unique information and it is not useful to compare these values with others. The only way to understand if the input image is the same (with the one among the database), is to look at the input image and reconstructed image, because when the input is the same on the reconstructed image part we are able to see the same input image. If the input image is not one of the database images, in which the case could be in real life, on the reconstructed part we are seeing a mean image which seems irrelevant with the input image.

Finally, it can be said that this program is not useful for recognizing the difference between human faces in different conditions, but it is useful when it is used to make clear if any image is a human face. Because maximum and minimum values give logical results to us when an animal face or another image, which does not belong to a human, are used. Thus, this program can only be used to understand if an image belongs to a human for our case.

4.2.2. BPN Based Face Recognition Program

In this part it is tested that if the system finds the right faces in different conditions such as various illumination, poses, scaling with little illumination change and expressions. These can be also called realistic conditions. For the serious illumination changes, the system sometimes catches the right image but the results are generally random and there are lots of mistakes. (a=different illumination, b=different angled pose (30 yaw degrees), c= with expression, f=yaw angle of 5 degrees, and g=10 percent increase in distance ratio between the face and camera with little illumination change)

Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image
1a	8	1b	1	1c	1	1f	1	1g	9
2a	35	2b	5	2c	2	2f	2	2g	11
3a	11	3b	2	3c	3	3f	3	3g	11
4a	8	4b	4	4c	4	4f	4	4g	8
5a	35	5b	5	5c	5	5f	5	5g	11
6a	8	6b	15	6c	6	6f	6	6g	8
7a	4	7b	24	7c	7	7f	7	7g	6
8a	8	8b	8	8c	8	8f	8	8g	8
9a	8	9b	4	9c	9	9f	9	9g	8
10a	4	10b	10	10c	10	10f	10	10g	6

Chart 4-4. The Results of the BPN Based Software for the Realistic Conditions

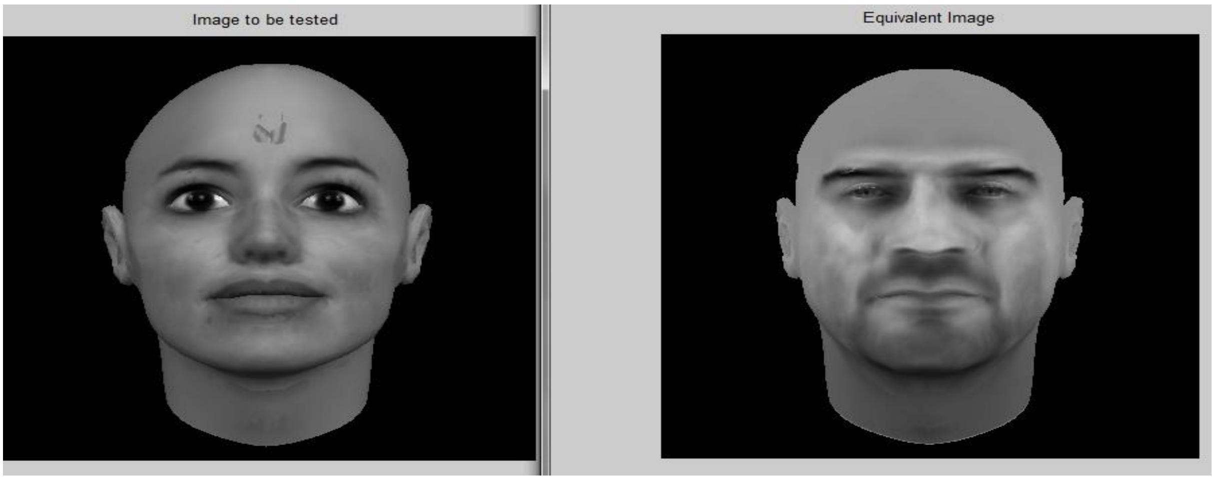


Figure 4-5. An Incorrect Matching by BPN Based Software

In case of the tests have been made for the images in small angle differences, we can say the program is very successful and able to catch the right person. Actually this rate is more than ninety percent in regard to databases, which can be considered as a great rate. It is always good to comment by thinking other possible databases since the a hundred percent success rate was valid for these tests made by me, because it does not mean the program will catch the right image for every single test. Since we are dealing with ten test images here, the success rate will be never be between 100 percent and 90 percent and just as this situation when it is tried to input more images to the system, the mistakes of the program might be evident. However, for our ten input images there was no mistake for small angle variation. The each underlined numeral shows a right match.

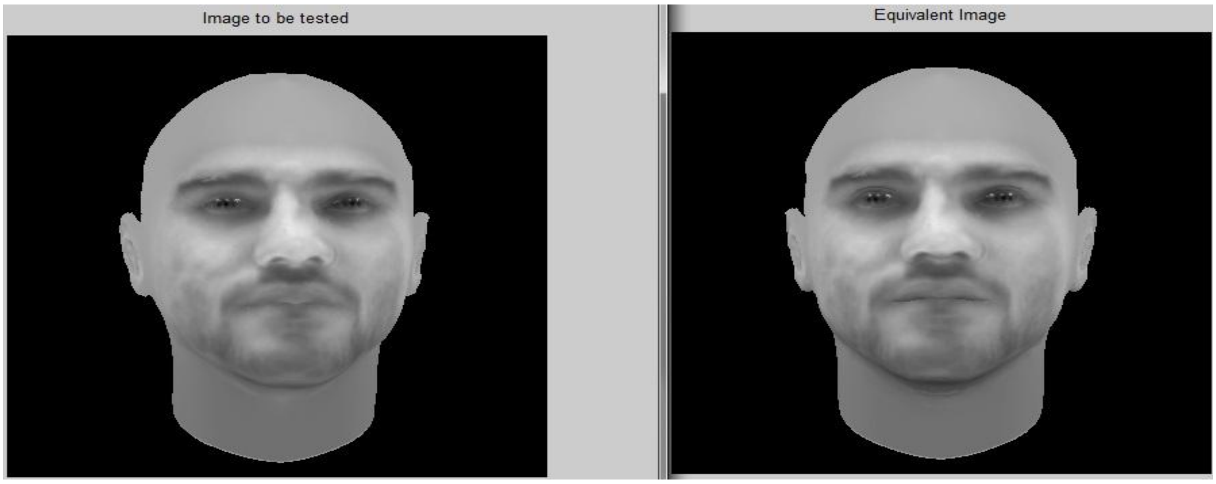


Figure 4-6. A Correct Matching by BPN Based Software

When have been made for the images in higher angles such as thirty degrees the program is unsuccessful. It only sometimes finds the correct image, but generally the found images did not belong to the same faces. So, software is not reliable when it comes to angle changes. In whatever way this cannot be seen as a deficiency of the program inasmuch as this angle is really big and the other softwares have more serious problems with this amount of angle variation. At least the system somehow manages to find the correct person, even if it is rare.

Figure 4-7. A Correct Result Showing an Interface of BPN Based Software

According to test results the system is again successful for the expression changes, which is another important problem in face recognition, and the all results seem right. Of course it does not mean that this success, for the each and every person in our test database, will be repeated for other test databases. But it is obvious that this rate of success will be at the vicinity of 90 percent and probably more than that. Hence, this system proves its success also for the expression changes.

The variation in scaling part “g” is the section where the failure of this program appears. In regard to results there is only one correct result and that is the eighth person’s test. But a reason exists for us that is hindering us to make this comment. When we look over the other results we see that the result of other three tests is also the same so it is better to think that result is a chance event. However, it would not have been an accomplishment anyway due to our expectations and the one correct result out of ten tests is obviously a bad result. Here it can be note that the results belong to four different people which can be taught as an evidence of the mistake of the program. These for people have darker skin (two of them are black) compared the others and this might be the reason of those results. Because when the distance between camera and the face increases, the portion of the black background on the image increases and this can be sensed as a black person by the software. In order to make this effect clear we can think an image that has a white background. In this case, the program would probably choose the ones among people who have lighter skin color. However, there is a more important result than these observations and it is the failure of the software concerning the distance ratio variations between the camera and the face.

4.2.3. Eigenface Face Recognition System by Amir Hossein Omidvarnia

It is always important to try the system in other conditions as well. To show the results better that is why it was thought that it is needed to be tried by other images which are the images from the exactly same faces following one after another, such as reduced brightness of illumination, different angle of the face (30 degree to the right), different expressions, different small angle of the face and finally scaling. This database is much more interesting than the first one, because in reality it is almost impossible to get the same images from the same face if the photos are taken in different time intervals. The training database is also set up with twenty different images that belong to various faces so that among twenty faces the ten same faces are used in the test database but with different conditions. The results found during the tests are like the following chart, where numbers represent the sequence of people and letters represent the different conditions.

Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image
1a	11	1b	18	1c	18	1f	18	1g	7
2a	13	2b	16	2c	16	2f	16	2g	1
3a	1	3b	16	3c	16	3f	16	3g	1
4a	4	4b	7	4c	7	4f	7	4g	6
5a	10	5b	16	5c	16	5f	16	5g	1
6a	15	6b	1	6c	1	6f	1	6g	6
7a	7	7b	17	7c	17	7f	17	7g	11
8a	4	8b	11	8c	11	8f	11	8g	8
9a	9	9b	1	9c	1	9f	1	9g	9
10a	7	10b	17	10c	17	10f	17	10g	11

Chart 4-5. Results of the Eigenface Based Software by A. H. Omidvarnia for the Realistic Conditions

In the chart the underlined values indicate the correct tests. In the first two columns are for the images taken in darker illumination conditions. There are just three tests that are successful out of ten and the system is not enough for the differing illumination conditions. The following pictures show the test for 8th person, but the system realized it as it is 4th person.

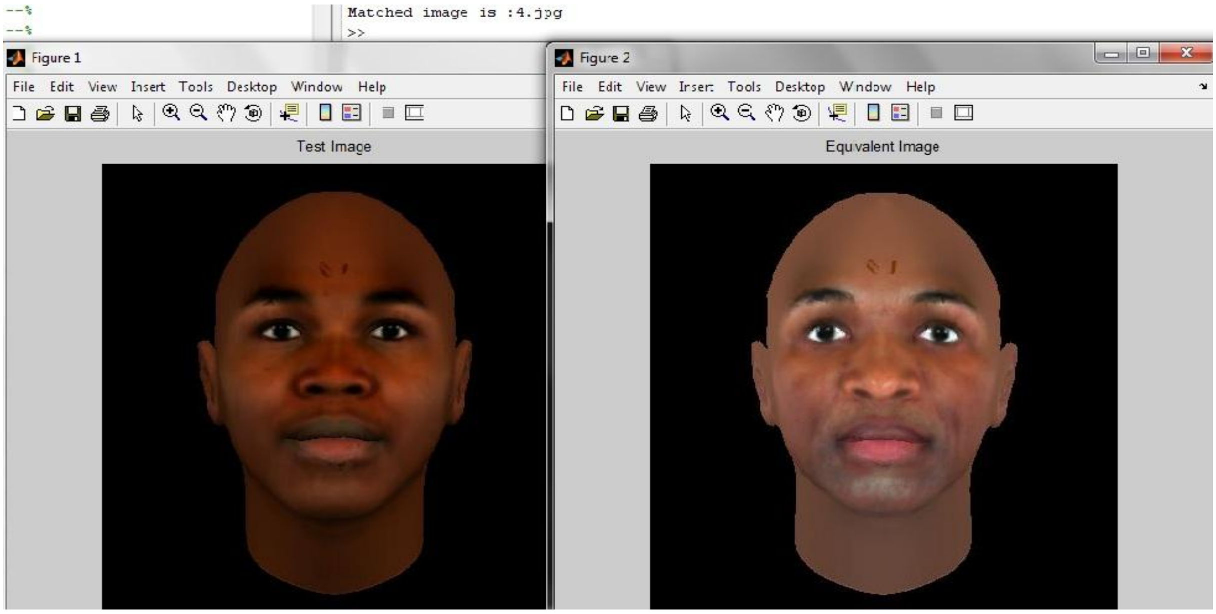


Figure 4-8. An Incorrect Result from Eigenface Based Software by A. H. Omidvarnia

The second pair of columns represents the tests under the different angle of the faces. The results are all random as can be seen. There is not even one correct result.

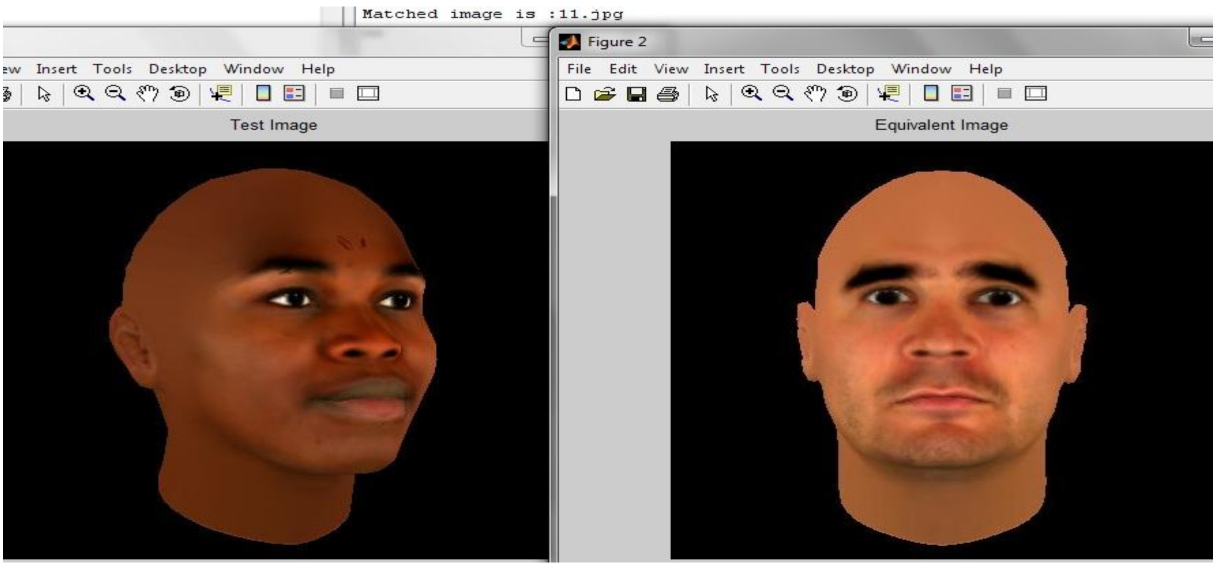


Figure 4-9. Another Incorrect Result from Eigenface Based Software by A. H. Omidvarnia

The image shows another photo of the 8th person and now the result is more erroneous. This time the equivalent image belongs to a white person while the 8th person is black. It can be noted that within the different expressions of people there is also no correct result and the same mistake is repeated again for the 8th person. Thus, we can say that this system might work well for the different illumination techniques but there is no possibility to find right faces within a huge database in case of the changes of poses and expressions.

The most remarkable success of this software is in distance ratio. When the distance between camera and the face is increased ten percent the system is still able to catch the right image sometimes. In our tests the right matching has been achieved three times and so has a rate of thirty percent. This rate might seem very low for a face recognition program but the affect of the so-called inconsiderable illumination also takes place here and the two types of variation that were used together can cause this result. However, since the softwares are sensitive to scaling changes, this is not a bad result.

4.2.4. Fisherface Face Recognition System by Amir Hossein Omidvarnia

After the good results for ideal conditions, better results can be expected from this system than the others. Here are the results of this program when it is applied to the same database.

Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image
1a	14	1b	4	1c	1	1f	10	1g	10
2a	2	2b	13	2c	19	2f	13	2g	10
3a	3	3b	16	3c	16	3f	10	3g	10
4a	4	4b	13	4c	4	4f	14	4g	10
5a	5	5b	13	5c	5	5f	5	5g	13
6a	6	6b	4	6c	6	6f	10	6g	10
7a	7	7b	4	7c	7	7f	14	7g	10
8a	8	8b	13	8c	8	8f	19	8g	6
9a	9	9b	4	9c	9	9f	9	9g	10
10a	3	10b	4	10c	10	10f	10	10g	10

Chart 4-6. Results of the Fisherface Based System by A. H. Omidvarnia for the Realistic Conditions

It can be noted again that for this designer's software we are using the first twenty images in the training database and that is the reason there is no result greater than twenty in the test results. In the end this condition will be reconsidered. However, the test images we are applying are always the same of course in order to get a fair comparison. In the first pair of columns there are the images of faces taken in a darker condition and the others are also same again as the previous tests. It is so easy to notice the difference than the previous one because of the abundance between the underlined numerals.

For the varied illumination there is an eighty percent success rate for this system according to this set of tests. The program makes errors for the 1st and 10th images for the first part but in whatever way it is still more successful than the previous Eigenface method. In the second part, the results change back again and the system is not able to catch anything even for once and just gives four different results for ten tests which ten different faces are used.

Figure 4-10. A Correct Result from Fisherface Based System by A. H. Omidvarnia for the Angle Change

In this case it is better the show the affect of angle and we can reduce the yaw angle and analyze the system again. When the system analyzed the yaw angle reduced to five degrees we can see the results on the last pair of columns (tests indicated with f). It is easy to observe that the results are better since the system is able to give the correct results sometimes as shown on the picture above.

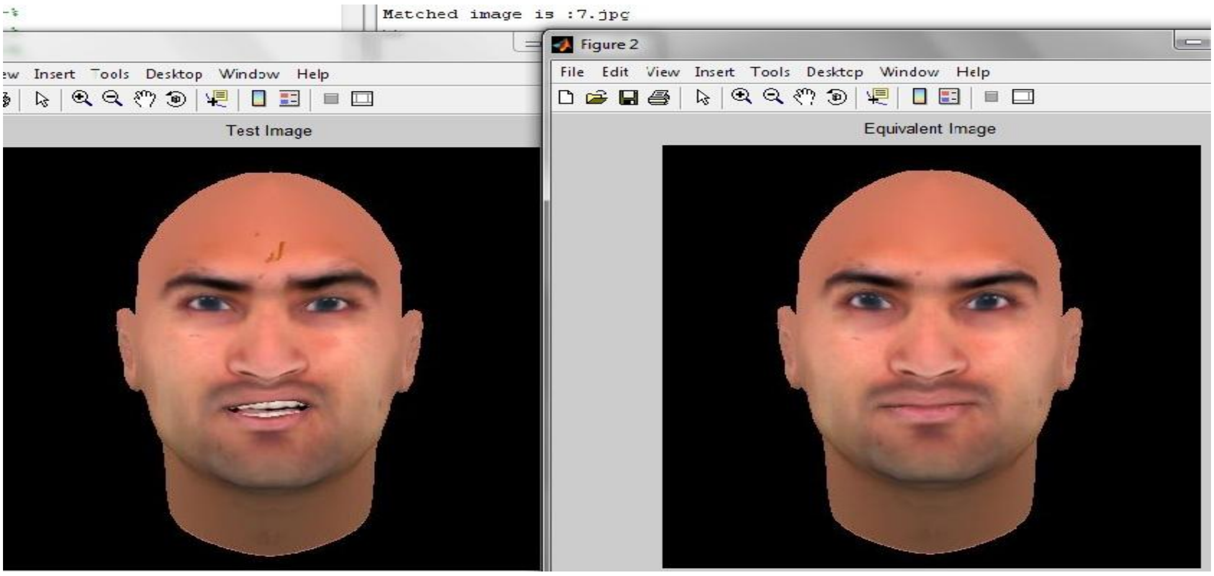


Figure 4-11. A Correct Result from Fisherface System by A. H. Omidvarnia for the Expression Change

The picture shows the change of expression, for the seventh test, and its success is as important as it was understood in the previous example since there was no good result but this time the Fisherface Method gives quite good results. The system's accomplishment rate is fair enough as compared to the previous Eigenface Method. There are two mistakes for the second and third faces but the eighty percent success is now achieved again for also the case of expression alteration.

For the cases of low angle and scaling variation the results are not good. In fact, for the small variation of angle (5 degrees) results are better than the ones that were achieved for the larger angles. This explains the hardness of identification when the angle gets wider. In the last part it can be seen that the correct matching was just caught in the tenth try, but since the program did tend to find the same result for other tests we again do not count this as a success. The other interesting fact for scaling variation tests is the "10" result has been found eight times in ten tests. This means that the program was stuck on the tenth person anyhow and actually after achieving first four same results it was not needed to keep on operating other tests since the other results were predictable and they had no meaning.

4.2.5. Face Recognition System by Zubair Saifi and Suvidha Sadhu

The correct results that were obtained in the ideal conditions aroused curiosity about the success of this program. The results would not be the same of course, but the in respect to my first prediction, there would be more than seventy percent correction at least for the light variation of conditions, which we can accept as a success among all other programs.

However it must have been tried under the change of conditions, which were also altered for the other softwares, in order to make a good comparison of the programs and to choose the best one among them.

Here are the results found for the program;

Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image	Test Image	EQ. Image
1a	NMF	1b	32	1c	1	1f	1	1g	9
2a	35	2b	5	2c	2	2f	2	2g	11
3a	11	3b	2	3c	3	3f	3	3g	11
4a	8	4b	4	4c	4	4f	4	4g	8
5a	35	5b	5	5c	5	5f	5	5g	11
6a	8	6b	15	6c	6	6f	6	6g	8
7a	4	7b	24	7c	7	7f	7	7g	6
8a	NMF	8b	8	8c	8	8f	8	8g	NMF
9a	8	9b	4	9c	9	9f	9	9g	8
10a	4	10b	10	10c	10	10f	10	10g	6

Chart 4-7. Results of the System by Z. Saifi and S. Sadhu for the Realistic Conditions

In regard to the results, we see a real inconsistency between the columns, which means the software is somehow sensitive for some of the changes such as illumination and scaling changes. However, the failure in the last column could have been resulted from the little illumination change again. Because for the much higher change of illumination, we see that the software could not get any correct matching even once for the tests which are stated in the first column. The “NMF” indicates that the software could not find any result for that test. For those cases the system says “no matches found!” thus from that point on it was shortened to “NMF” to supply a better chart of results. This feature is interesting as well because some of the programs do not have this result in their source codes. This might be both good and bad for different cases. Here the software said that for the tests conducted for the first and eighth person,”no matches found” and it is a real problem that the system could not match any of the images with each other within the database. Because this one was going to be the result of us if we inputted an image from the outside of the database to the program, thus we can accept this result as a program mistake.

When it comes to variation in angle, the system found four correct results, which is not so bad because the change of angle was thirty degrees. The results have been obtained during five degrees of variation are better, which corresponds to one hundred percent and we can say that the system works satisfactorily in case of angle variations.

The expression changes are absolutely the easiest variation kind for this software. Whatever the expression was the system got the right face within the database so that is why all the tests are underlined for its column. This part is the only part the system reached a hundred percent success among our tests.

Figure 4-12. A Correct Result from the System of Z. Saifi and S. Sadhu for the Expression Variation

As to scaling changes there is no success for any of the tests and once the software could not match the right image. So now it seems that the system is not good for variation in scaling or maybe this rate of variation (ten percent) is too much for all softwares.

5. Analysis and Comparison of the Results

In order to understand the performances of approaches and softwares, in the beginning it is better to analyze them in both ideal and real conditions. The softwares were tested here in this project were the most promising and reliable ones, thus the results expectations were high. The results in ideal conditions are generally perfect. However, it is hard to catch the same success in realistic conditions; therefore, the success rate dropped to even zero sometimes from ninety or a hundred percent. It should be noted that there can be small mistakes for the software results, which can be caused by any of a lot of aspects which take place in face recognition. And it can be because of the test amount since the more test repetition makes better results and sharper success rate possible. For instance, in the ideal case, the reason of one mistake in BPN based approach but the entire successes under the realistic conditions are because of this fact. Otherwise, for a program, it is not possible to be more successful in realistic conditions than in ideal conditions. So this situation was simply caused by the other more random tests that have been made in ideal conditions. However, for the tests have been made in realistic conditions, there were a lower number of tests since a lot of different conditions were tested. This means, in realistic conditions, a hundred percent success ratio can be lowered if we conduct more than ten tests. However, we try to understand the performances better and in following paragraphs the figure of merit used is the percentage of correct recognition in the real cases. It can also be recalled that for softwares of A. H. Omidvarnia, there are twenty images in the training database instead of forty like the others in order to make the tests more clear, because in case of forty images in the training database, the softwares are more liable to possess mistakes and sometimes it is so hard to get results. This situation, which will be better understood in later sections, will not affect our final comparison and results.

Generally, in the ideal conditions all the softwares' results are good. The reason for that is the usage of the same images in both and training databases. The ideal conditions are the fundamental part of face recognition. In this case we only input the images which are also present in the training databases. As we use one of the images in the training database, the systems catch its corresponding image in training database which is exactly the same for the ideal case. Thus, for this case it is better to say that the systems are not matching the same faces but they are matching the same images to one another. Since the images used are the same in both training and test databases, it is not a big deal for a program to find their matches. We see the real problem begins when the faces have expressions or are under the different conditions mentioned before.

Among the carefully selected different softwares the first one that was made in Drexel University has a slightly different purpose than the others, but since it has good steps that are useful for understanding the general idea well, it was worth to analyze it carefully and put the results of it.

The program has a difference from the all others because it shows the Euclidean distances. This Euclidean distance lets us know how close the input image is to the other images in the training database. Then we are able to comment about the tests by analyzing the maximum and minimum values and we can say if the image is a known face (it is already in the database), an unknown face, or not a face. But here we should note that it does not show us the matched image from the database and this is where we see the difference of this software than the others. Thus, the program is about using the known face as an input then observing the results and then using another face that is not known whether it is from the database or not and understanding if it is from the database by comparing the results with the one's from the training set. Hence, for this software we simply decide dependent upon min and max distances. Like these results, we can analyze the cases of other images that do not belong to human beings. For those cases the results will be far out and we can discover that they are not faces. The steps of the program are well written and very useful for its purpose and enables us to understand the subject well, but, due to requirements of extra operations such as collecting results and making comments about them by ourselves, it is not good as other programs and not relevant enough with our exact subject, which is personal identification via first recognition. Finally, this program would not be the one selected anyway since it does not fit in with our interest and yet the program had problems in realistic conditions and also even in the ideal conditions in our tests made by using the new database that was produced by Facegen. The results for some of the faces did not fit the chosen criterion stated above and so this software is better used for its purpose but not for ours.

One of our important software based on BPN (Back-Propagation Neural Network) is better than previous one because we are able to see the image that is supposed to be corresponding one of the test images. In ideal conditions the system found the right faces except just for once. The first outstanding disadvantage of this program is the time. When we run the program the system starts to do iteration and determines the MSE (Mean Square Error) for each and every iteration result. Then after some minutes (may differ by the power of the computer) system concludes this part and asks the folder of test images. And this part really takes time when we would like to test another image and the program repeats these initial operations over and over again.

As for the tests in realistic conditions the results are really good. In the first tests there were doubts about the program because of its failure concern with the images that are achieved in weaker illumination because the system got the right result for only once and as stated before this cannot be counted as a good result. So we can say that this program's weak spot is illumination. This weak point was also realized when the tests have been made in slightly weaker illumination, and, for that case, the system could not pass over ten percent success rate. However, this might have also been because of the scaling change by adjusting the difference ratio between the camera and the face.

When it comes to advantages of this software and also the approach, we can say that the system works very well in angle changes of the face. It reached a fifty percent success rate for a wide-angle change, which was thirty degrees. Since this angle is big, it was no surprise that we would have had a better success rate for the five degrees of angle alteration in respect to original conditions. And finally, the same success repeated for also the five degrees of change and system reached the top. It can be also noted that, under this condition, this a hundred percent success rate is the best one among the other programs tested. Expression change is also not a big deal for this software and it reaches a hundred percent success rate again. Thus, this software is very good for every databases and it has a unique superiority in expression and angle changes. The only disadvantages are about the time necessity and the bad performance under the different illuminations. But the advantages of the program make this software indispensable for realistic applications.

Another program has been tested for both ideal and realistic conditions were the Eigen Face Recognition System, which was programmed by Amir Hossein Omidvarnia. In this program there is no need for grayscale images because we can use multicolored images. If we start talking about the tests, the results are not good. Actually there is lack of success for even ideal conditions and system usually gives wrong results. It was seen that, for a black female test person the result was a white black male and these sorts of mistakes cannot be tolerated in face recognition processes. After these kinds of mistakes it would be another mistake to hope too much from this program for the real conditions. The test results have not misled us and under the two important conditions the system could not even do one correct matching. These were angle variation and expression which are so important in personal identification.

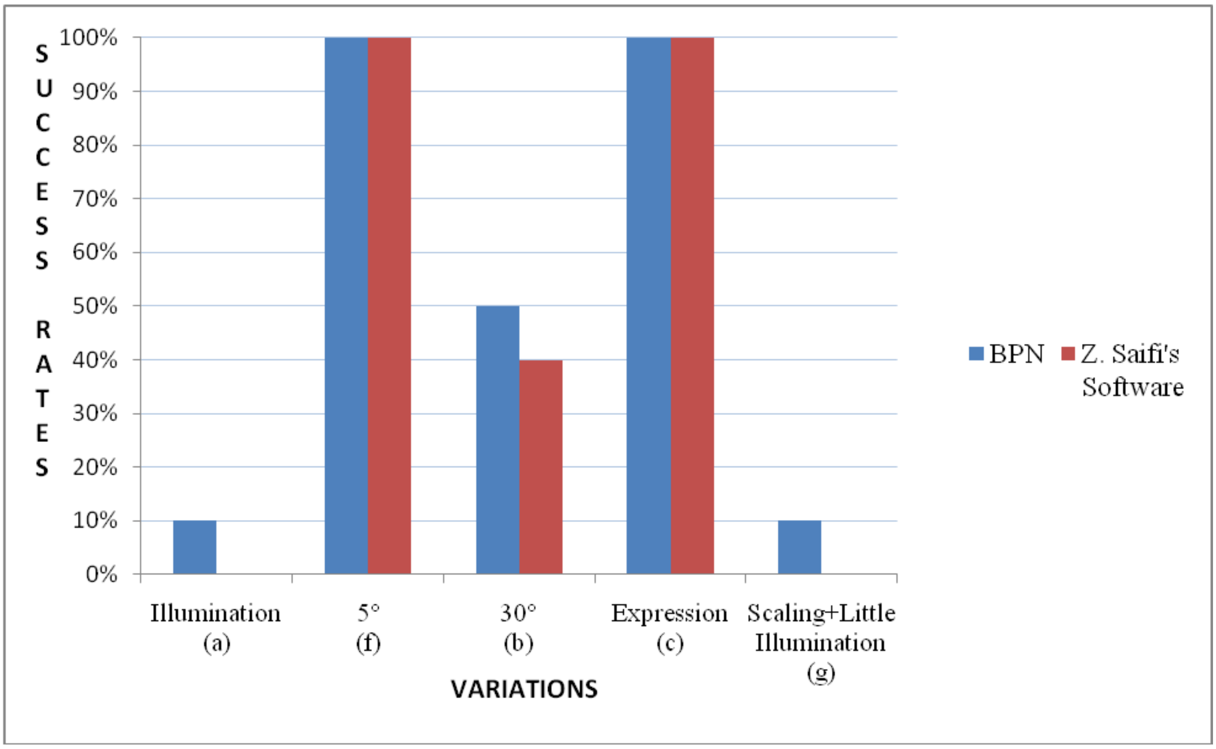
However, for illumination and scaling alterations, the system was not affected too much and found thirty percent of images correctly. These rates might appear to be very low but especially for the ten percent of distance ratio change (increase in distance) the thirty percent of success is not low and actually this is the best rate that has been achieved by the tested softwares. Nevertheless, it is good to be cautious when we have to comment about these cases because since the software, contrary to some programs, tends to find a result all the time and this result can be random so they can be achieved by chance. For even a program which is no longer in the right working order these results can be achieved by chance; however, we are supposed to ignore the chance factor since we are talking about the test results and we should be conscious of what happens if we add also that factor because probably there will be no way to make reasonable remarks in that case. In the end the program would never be in the vicinity of our choice, if we were to choose one of them, due to very bad performance in expression and angle variations. So we can say this program, unfortunately, fails.

Then our next program was Fisherface Face Recognition System by the same programmer, Amir Hossein Omidvarnia. It is better to denote that this system was the modification of the prior system, which was based on Eigenface algorithm. The system aims to have better results by the adaptation of fisher algorithm and reaches its goal. In the ideal conditions we can see this success since the system did not make any mistakes in our tests. This is the first advantage of this system if it is compared to the previous one and this was the signal of later success in realistic conditions.

For our first realistic condition the result is very good. The software reached an eighty percent success rate for the illumination and this result is the best amongst the rest so it can be said that this software has great advantage under the weaker illumination (for example the condition in case of a darker room) compared to the others. When it came to thirty degrees change of the angle, the system was not able to do better than its prototype and it ended up the tests with zero success. Also for the five degrees of yaw angle change the system got a thirty percent of success which is not good compared to others. Therefore, we can say this is a disadvantage of this software. As to expression variations, the system reached seventy percent which can be counted as a good result and in this point it is obvious that the modification worked well. Because in any case the modified one is far beyond than its prototype except in case of scaling changes where the modified one got the right result just for once. It seems that after a good beginning with the illumination conditions the system could not do better than the BPN based program and the variation of angle can be crucial in personal identification so this disadvantage unfortunately overshadows the success in illumination variation for this software.

The last software which has been tested is Face Recognition system that uses Eigenface approach and it was programmed by Zubair Saifi and Suvidha Sadhu. The first salient difference between others is a proper menu and the first disadvantage is the time needed for calculating the eigenfaces however it does not have to be repeated like BPN based software. But if we enter the wrong key during the inputting the test images, the program ends the session since it was programmed in that way (as a menu function “pressing any other key” means “quit”) and then we have to start the program over again,, which means to wait for the Eigenface calculation to take place again. However, this disadvantage can be counted as irrelevant one.

The tests made in ideal conditions resulted in successfully. The success rate was a hundred percent which is the same with previous software’s result. However the results obtained in realistic conditions are always more important for us since they would be closer to real results that can obtained by practical tests or even real examples of these applications of the present time and the future. So, under the real conditions, the system started with a bad result that meant nothing for the change of illumination. When it was tested with the faces that were looking towards right with an angle of thirty degrees, a good result was achieved by the system. For this case the success rate was forty percent and this was almost the same we had from the BPN based software so both of them are sharing the first place in this respect among the softwares tested. If we keep on analyzing the software we see that there is another good result for the expression variation case. The result is a hundred percent and this is same with the BPN based software again. So the “a”, “b”, and “c” did not give us a clue about which software is better (except a small difference in illumination variation). However for a smaller angle change we can compare the results again and it can be noted that the BPN had achieved an entire success. But we see that this value is again a hundred percent for our last software. This good result cannot be enough to select this program for us as the best one (also reminding the small difference for the wide angle variation) because for scaling we also see there is no correct result among the tests.



Schema 5-1. General Performance Comparison of the Best Softwares for the Realistic Conditions

After gathering all results from the tests in ideal and realistic conditions for the programs and by remembering our figure of merit we can now decide which one has the best algorithm up to here. Between the programs of Amir Hossein Omidvarnia, the first one was unsuccessful but the second one was good at illumination variations in spite of that the training database was the half compared the one used for other softwares. However due to lack of performance in expression and angle variation cases we cannot choose this program as the best one. As to the last software, results were too good but since the BPN based software's results are slightly better, up to here, we can say that (by also commenting about the graph) the best software was BPN based one which uses Back-Propagation Algorithm but the softwares are going to be compared again in later sections. As another aspect, it can be also noted that the Fisherface was far better than Eigenface version because even for the same program but for change of those algorithm resulted perfectly and we caught a better success rate with Fisherface based one. However it is still early to define the best software at this point because we will perform more tests in later sections, but these results were for the tests have been made until here.

6. Detailed Analysis of the Best Algorithms

In this chapter the results of new tests are considered for the best softwares in order to find our best software remembering our figure of merit. For all of these softwares, the database prepared by me was used again. The generated Facegen faces are always ready to be used for these cases. After the faces are completed, they can be reached anytime and it is possible to make changes to them again. So from the beginning to the end this database has been used, which has forty people in it, and it has always been used without changing the people but changing the conditions stated many times before. In this chapter, there are two tests that are being operated for ten people included in the database for each software and the results are clear to see how the software is to work in realistic conditions. Thus the aim of this part is to test ten people with the database changing the scaling, which means the distance between the camera and the 3D face model here, and also the yaw angle meaning the angle of view.

It is better to pay attention to scaling changes made in previous chapters was different than this one, since that change was done also under the different illumination condition which was very little but should have been effective so the same results cannot be expected which were had in the previously made tests under the ten percent of scaling variation that corresponds %90 (ten percent decrease in scaling) here. Moreover, for the Fisherface based software now it is time to use forty-imaged training database for the fair comparison and see the effects, yet after that part, due to very bad results (which will be further explained later) a new pair of tests will be made in order to understand the program's response and to analyze it independently.

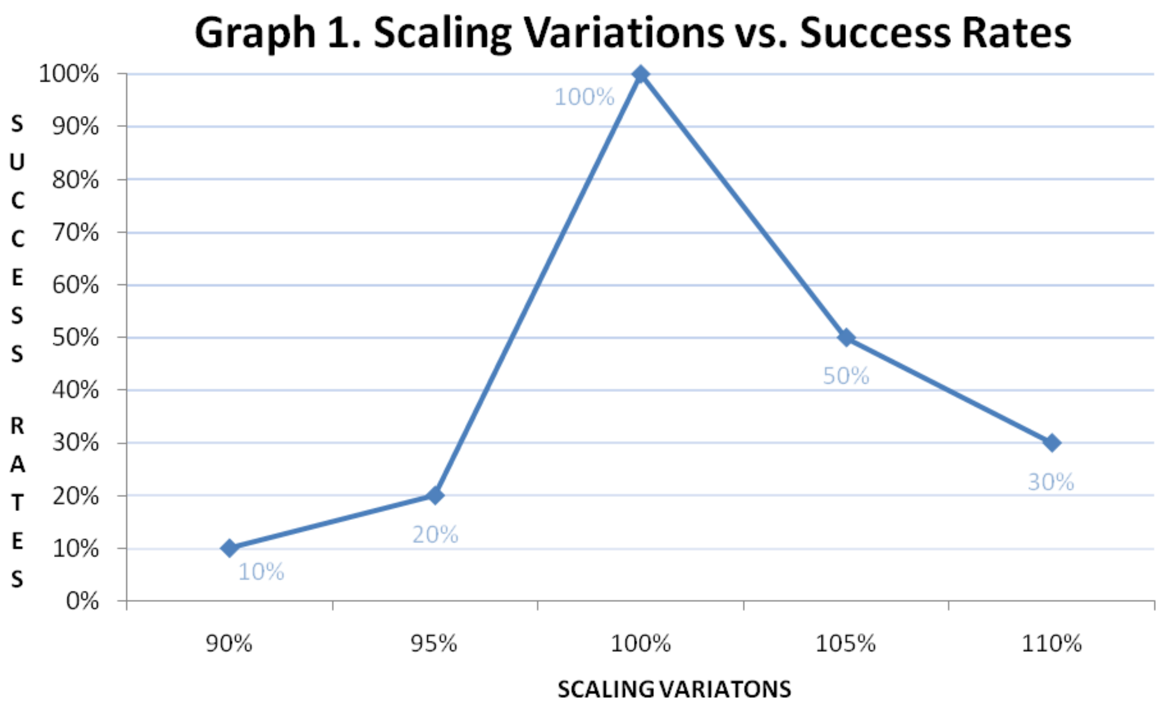
6.1 BPN (Back-propagation neural network) based Face Recognition Program

Here BPN based software, which was our best software according to the previous results, is going to be tested again as well as other good softwares in order to find our real best software. And we start with the change in scaling;

The Effect of Scaling of the Image Size to BPN System					
	90%	95%	100%	105%	110%
1	11	20	1	32	13
2	20	1	2	5	5
3	20	1	3	33	22
4	8	6	4	4	4
5	20	34	5	5	5
6	8	6	6	9	15
7	15	11	7	24	24
8	8	8	8	8	8
9	8	6	9	9	35
10	11	7	10	10	13

Chart 6-1. The Effects of Scaling Change to BPN Based System

Here we see that for the first column of %90 scaling, the results found before with the same variation of scaling plus also the presence of difference illumination conditions are different, but in the end the success ratio is the same, which is ten percent. Just one correct result out of ten tests may seem as a bad result but the bad effect of scaling changes are undeniable in face recognition, yet a failure in just one of the conditions does not mean the algorithm of the program is not proper since it is really hard to find a software which reflects well to every kind of variation. As to second column of our test, it does not seem suitable again. The result is twenty percent success, meaning failure for this distance between camera and face, which is not good for our software to recognize the faces properly. With these two columns we reach a result already that is when the images are in a small size than the ones used in the training set, the problem begins in personal identification. In other words, when the distance between the 3D face model and the camera increases the system gets erroneous.



Graph 6-1. Scaling Variation Effects for BPN Based Software

In the third column there is no need for so many comments because those tests are about the ideal conditions and they are useful only since they show the software reaches top at that case. Hence in ideal cases we are seeing perfect results. Ideal tests are the cases where there is no change in conditions, and here the ideal case enables us to make a good comparison between the performances. After reaching this high it is not surprise for us to drop to the same level again, but it is important where we will experience that situation. When we decrease the distance ratio more (meaning increasing the size of the image) we see that the success ratio reduces all of a sudden. However, this ratio is up to fifty percent, which is still more successful than the one made in the opposite direction. This is the first evidence of the sensitivity of this program against smaller sizes. Then we decrease the ratio with the same rate as can be seen in the graph. In other words, we increase the sizes of the images and then the success rate keeps on diminishing up to thirty percent. Thus the system was able to match the right images three times and this is also better than the ones made under the %90 and %95 scaling conditions.

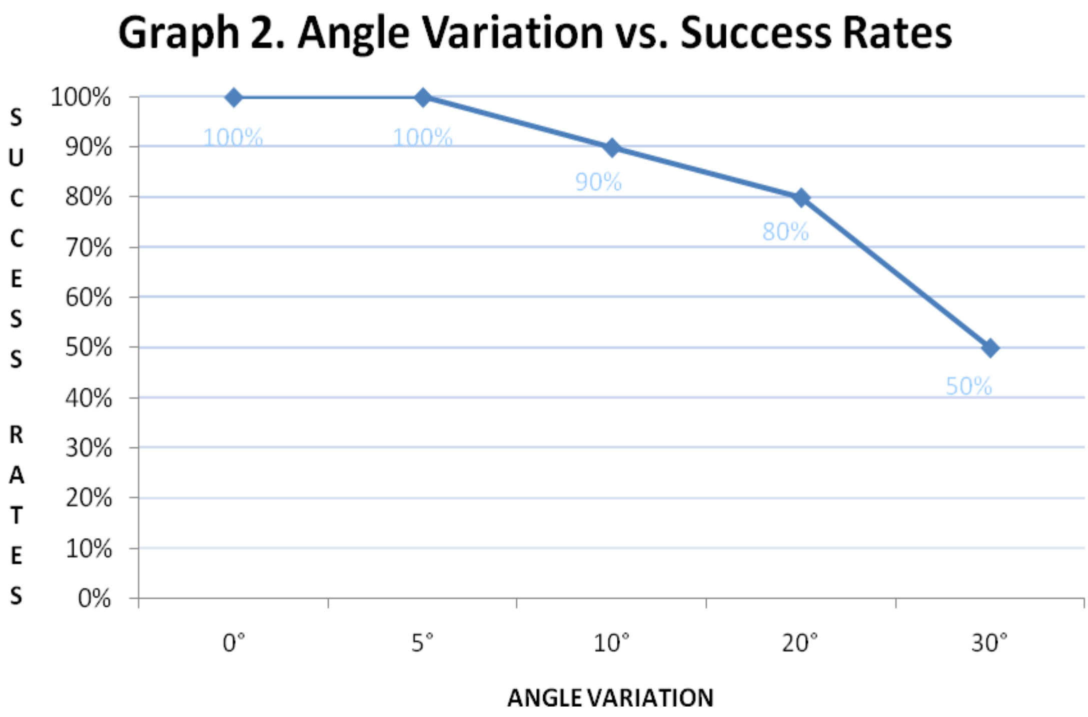
The second significant group of tests shows the effect of the yaw angle variation. Here, some angles are tested from zero to thirty degrees and the success in each different angle can be shown as;

The Effect of Yaw Angle Variation to the BPN System					
	0°	5°	10°	20°	30°
1	1	1	1	1	1
2	2	2	12	5	5
3	3	3	3	3	2
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	9	15
7	7	7	7	7	24
8	8	8	8	8	8
9	9	9	9	9	4
10	10	10	10	10	10

Chart 6-2. The Effects of Yaw Angle Change to BPN Based System

According the results the software seems successful for matching the correct faces in different angles. As to first one there is no angle change made and thus the results are all correct as it is representing the ideal condition. For the five degrees of change what we see is the result is same and the software is not affected at all from that small angle variation. As we increase the angle the errors will start.

For the case of ten degrees the software made one mistake according to the results. This is not bad and can be considered as success. This case might be seen in real case more that the bigger angle since it is hard to fix someone's head to just frontal direction during the taking of the photo. It is noticed that as we increase the angle the success rate decreases as in the case of increasing the angle from ten to twenty. For this case the software did another mistake and the success rate is for this angle variation is eighty percent. The last case was tested in the one of the previous chapters and since the conditions are equal here, the results should be exactly same. However, the software was run again to avoid any mistake but the results that were found before have been gotten again and this lead to a dramatically change in the success rate which is now lowered down to fifty percent. If we increased the angle more we would notice more decrease in the success rate. These results finally enable us to see that fact on a proper graph.



Graph 6-2. Angle Variation Effects for BPN Based Software

6.2 Fisherface Face Recognition System by Amir Hossein Omidvarnia

In this section, first of all, the Fisherface face recognition takes place with forty images in the training database. It can be noted that this situation was not same for the previous sections where the database of training images was composed of twenty images. This is due to the fact that the original of this program was prepared in that way and when this training database size increases the results begin to become distant from the correct ones, as it can be seen from the following results.

The Effect of Yaw Angle Variation to the Fisherface Face Recognition System					
	0°	5°	10°	20°	30°
1	6	6	6	10	39
2	40	40	39	1	1
3	17	33	9	17	10
4	17	6	37	36	29
5	33	25	6	1	1
6	17	17	10	39	40
7	14	14	36	1	35
8	14	21	14	36	35
9	18	9	6	6	6
10	18	10	6	6	1

Chart 6-3. The Effects of Yaw Angle Change to the Fisherface System

The table shows the effect of yaw angle variation to the Fisherface face recognition system. It should not be forgotten that these results were achieved when the training database had forty images like in the case of BPN based program and the face recognition system of

Zubair Saifi and Suvidha Sadhu which takes part after this section. Thus this part will be the compared part with other two softwares in the end to understand which one is better.

As to results, the mistakes are very common and actually there are just two correct results. Since they are not in the column of zero degree indicating ideal conditions, we cannot count those as real right results. They were obviously found by chance because there are no right results for the ideal tests. In the ideal conditions there are ten results, but it is interesting that only in the one of the results they found image is within the test images which is 6th one (but still it is wrong because the result should has been "1") and in the others there are far out results. This failure resumes from beginning to end and it can be simply said that for angle variations this software is a disaster for huge training databases. Following chart shows the scaling change effect to this program's success and at first sight it seems that the failure experienced for the angle changes repeats also for this case.

The Effect of Scaling Variation to the Fisherface Face Recognition System					
	90%	95%	100%	105%	110%
1	18	18	6	39	6
2	25	17	40	39	39
3	18	33	17	6	25
4	33	33	17	6	39
5	18	18	33	34	10
6	18	18	17	17	6
7	18	33	14	13	22
8	39	5	14	14	14
9	18	18	18	40	6
10	18	17	18	34	25

Chart 6-4. The Effects of Scaling Change to the Fisherface System

For our database, which was formed with forty images, the tests have been done for the scaling variations resulted with failure and this states the Fisherface Program is good for nothing when a training database formed by forty images is used. It should be again noted that the original condition of the software was with less training images and for that case it responds very well for some conditions. However, in this last part the three programs are tested with same conditions where the training database has forty images and thus in this comparison the system has no success and cannot be thought as best one among the others. There is only one correct result seen on the last column indicating ten percent greater size of the pictures, which belongs to the sixth image, but one correct result does not reflect any success here. According to these results, the program is not good for huge training databases. Moreover, there is no need to show the results on graph since for both graphs there would be just two peaks in total and all the other lines would be adjacent to the x axis indicating the success rate is zero. In addition to that, a few correct results do not actually show real success rates of the program because for some of the columns the rates are around ten or twenty percent, but all the others are zero, meaning that the correct ones were found by chance and system results haphazardly. For this reason there is no need to compare this program with others and since the Eigenface System by the same designer, Amir Hossein Omidvarnia, was more unsuccessful, we can say that the designer's both softwares are not good to be compared with the other two. However, for better understanding this Fisherface method by itself and separately, the same tests (but this time with less training images) can be made and the results can be analyzed.

Following chart shows the tests made with the faces when they have different yaw angle as before but this time there were only ten images being used in the training database instead of forty. Thus, the idea here is to understand whether the software can find the right faces, and, if it can, how it does with the less training images.

The Effect of Yaw Angle Variation to the Fisherface Face Recognition System(Training Database of 10 Images)					
	0°	5°	10°	20°	30°
1	1	1	1	10	1
2	2	1	6	1	1
3	3	1	10	10	5
4	4	4	8	8	8
5	5	5	5	5	5
6	6	1	10	10	10
7	7	8	10	10	8
8	8	8	8	8	8
9	9	9	10	10	8
10	10	10	10	10	10

Chart 6-5. The Effects of Yaw Angle Change to the Fisherface System for a Smaller Database

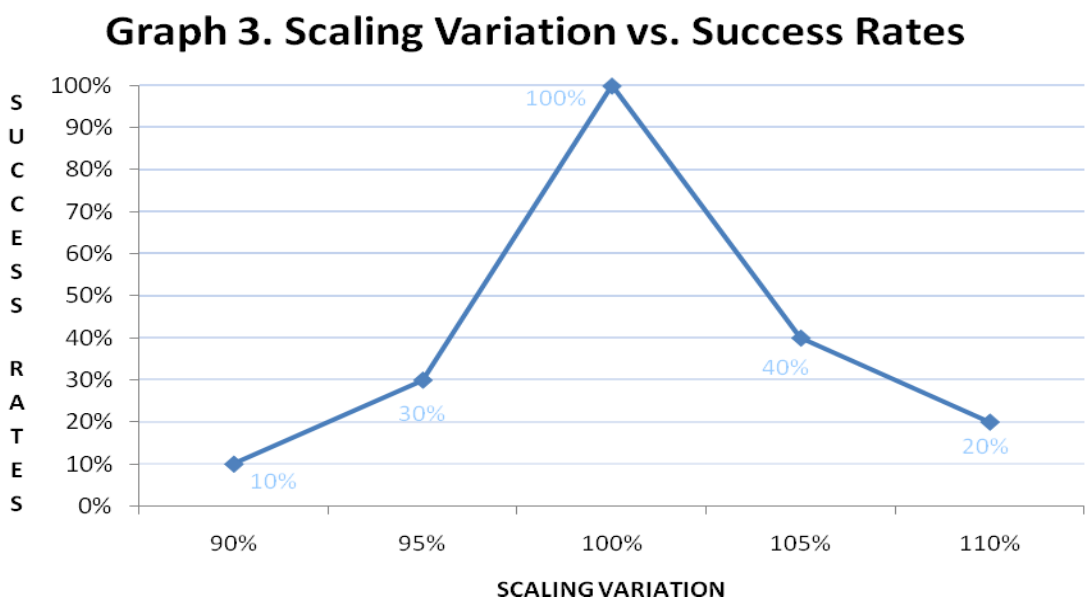
When we analyze the first column of the chart regarding the angle variation we see that in the ideal case the system has a perfect success rate. When we change the angle for five degrees, it results a decrease in success rate and moves it down to sixty percent. Ten degrees and thirty degrees of changes make the same difference and reduce the success rates more and drop it to forty percent. For the tests made for twenty degrees the results are more or less the same but a bit lower where we see thirty percent of success. So, this extra test shows us that it gets harder for any type of program to recognize faces when the yaw angle increases. This case is also visible also on the graph where it can be observed that the success rate drops down dramatically even after the first small angle change. It is known that this program is not enough for large databases, but it can also be said that now, for small databases, this program is very sensitive to yaw angle changes. Also, in the following chart, the performance of the program under scaling variations can be well understood for a limited training database.

The Effect of Scaling Variation to the Fisherface Face Recognition System(Training Database of 10 Images)					
	90%	95%	100%	105%	110%
1	1	9	1	1	1
2	1	1	2	5	5
3	9	9	3	1	5
4	10	10	4	4	3
5	4	3	5	5	5
6	4	6	6	9	1
7	10	10	7	8	8
8	10	8	8	3	3
9	10	10	9	1	3
10	9	10	10	10	8

Chart 6-6. The Effects of Scaling Change to the Fisherface System for a Smaller Database

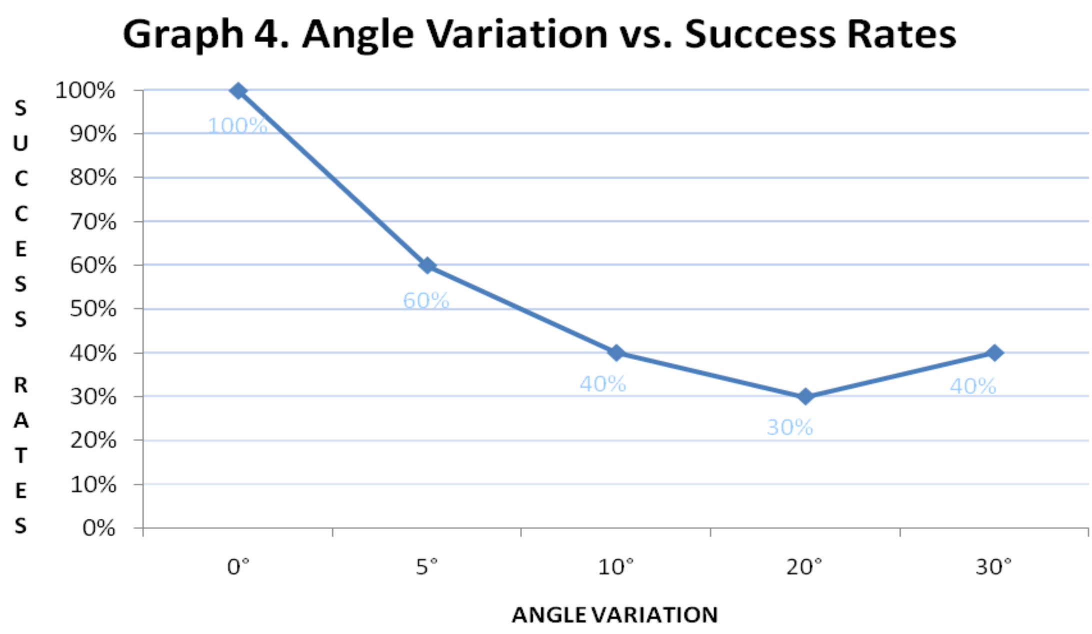
Here we see that in the first column, which can be counted as the hardest condition to be analyzed for any face recognition software, we do not see any performance from the software and the success rate is only at ten percent. The reason for this difficulty for the face recognition programs is because of the face images which are becoming smaller by size, due to increasing distance between camera and the face, because this leads a complexity in details where clarity is lost and background starts to play a more important role in face recognition. This case causes the software to catch the faces of wrong people; especially people with darker skin such as black people's faces are given as results by the software considering the background color is black. Here, the black color at the back might be causing false sensation of the images and it might be detected as also the color of the face since it changes mean color of the images making them darker, and thus a white human face plus a wider dark back ground is generally sensed by a black person's face. Finally, it can be said that, in these tests, sometimes it is possible to see this kind of mistakes which arise from the greater distance between camera and the face so these results, in the first column, conclude their mission that is important in order to understand the possible reasons of the mistakes.

As to second column meaning ninety-five percent of camera-face distance variation, we see a better ratio that is around thirty percent. For the third column, it is for sure that the results are supposed to be all okay since in the previous chart, the zero degree meaning the ideal condition had a full success and there is no difference between two columns since nothing has changed. When we increase the size of the place which is occupied by the face in the whole image, in other words when we bring the camera closer and thus when the face gets bigger in the image, it seems that the result is getting better than the opposite. However, the last column shows us that it does not last too much and after a limit the success rate lowers again as the face gets bigger and this condition generally repeats for all softwares. Thus, the results, which would not have been used for the final comparison of the three softwares, can be used for commenting the general performance of the softwares under the varying conditions at last as useful information. That is why it is better to draw two graphs for the last two tests instead of the ones with forty-imaged training databases and yet there would be no meaning of the graphs for the forty imaged ones since there was not any proper result.



Graph 6-3. Scaling Variation Effects for the Fisherface System

In the previous graph we see a peak in the middle of course and the graph lines show the previously stated effect well which is an increase in success rate as the face area increases in the image up to some limit. Like scaling variations, the graph of angle variation can also be drawn with the test results.



Graph 6-4. Angle Variation Effects for the Fisherface System

Here, it can be noted that there is a decrease in success rate from beginning to end except the thirty degrees one where the success rate reaches forty percent again. This instability could have been caused by the chance factor because when we use ten images in training database, even for any software which is not working right, there is a possibility of at least ten percent for catching chance of the right image when we input a test image. This chance rate would be just two and a half percent when we use forty images in the training database like before. Thus it is always better to use more images in the training database in order to understand the real efficiency of a program. However, it is proven that this program is not as good as the other two programss.

6.3 Face Recognition System by Zubair Saifi and Suvidha Sadhu

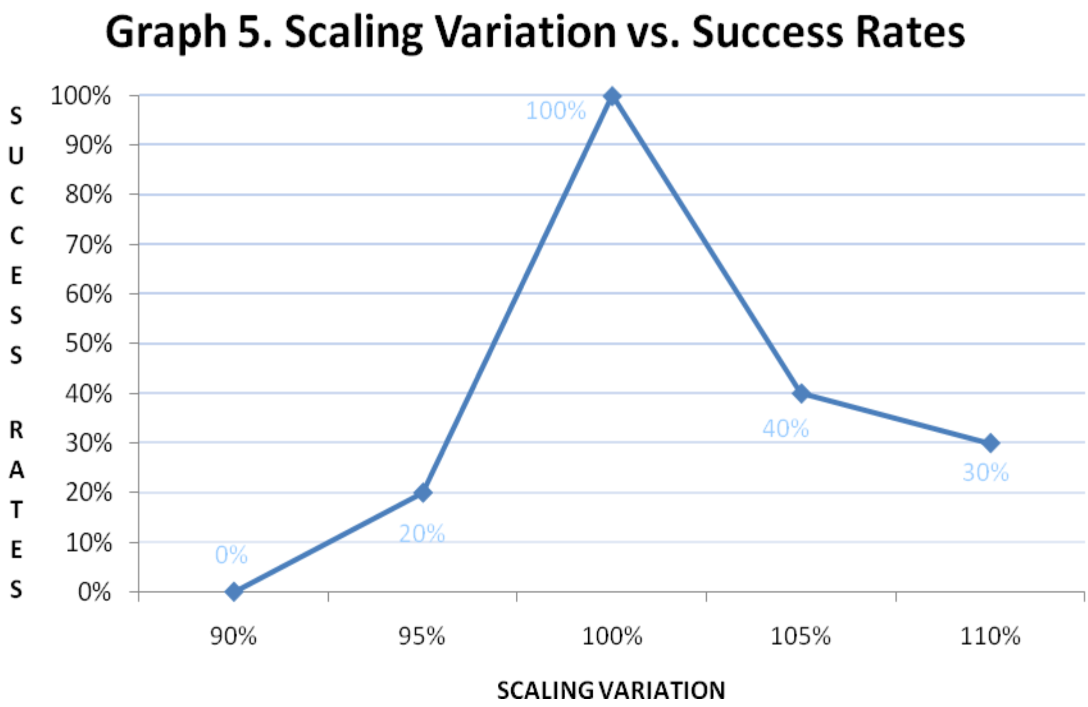
In this section, the new test results are taking part for the system of Z. Saifi and S. Sadhu. According to general results where the other variables, such as expression or illumination, took place this software was working very well and it was almost as good as BPN based one. So now we can understand which software is really the best one by enlarging the test results for this software as well.

In this case the same database should be used again for a fair comparison. Thus we can begin with the scaling variation;

The Effect of Scaling Variation to the Face Recognition System by Z. Saifi&S. Sadhu					
	90%	95%	100%	105%	110%
1	11	20	<u>1</u>	32	13
2	20	32	<u>2</u>	<u>2</u>	5
3	20	30	<u>3</u>	22	22
4	NMF	6	<u>4</u>	<u>4</u>	<u>4</u>
5	20	32	<u>5</u>	<u>5</u>	<u>5</u>
6	NMF	<u>6</u>	<u>6</u>	15	15
7	15	11	<u>7</u>	24	24
8	NMF	<u>8</u>	<u>8</u>	<u>8</u>	<u>8</u>
9	NMF	6	<u>9</u>	15	35
10	11	7	<u>10</u>	13	13

Chart 6-7. The Effects of Scaling Change to the System of Z. Saifi and S. Sadhu

What we see from the results of first tests is that there is no correct result. Moreover, for some of the input images such as fourth, sixth, eighth and ninth ones, no output image could have been found by the program and this case is shown as "NMF" meaning "no matches found" on the chart. There is another interesting fact alongside the failure here, and that is this function of the program. Because in other programs the sourcecodes tend to find outputs anyhow even they are not a reflection of the input. They are simply finding the results every time, perhaps they find the nearest images according to themselves. But this program does not do the same which means the chance rate previously mentioned is not as high as the others. When we look up the second column, we see there is twenty percent of success rate but there is no result such as "NMF" and we will never find that result again in these tests. So if we want to comment on this fact, this was happening because of the wider background where the face covers a smaller part of the image. It can be recalled that in other softwares, for this kind of cases, generally the results used to become black faces since the softwares sense the background as a skin color indirectly. For this software, in a similar but also as an opposite way we get "NMF" results when we test people faces with darker skin color.



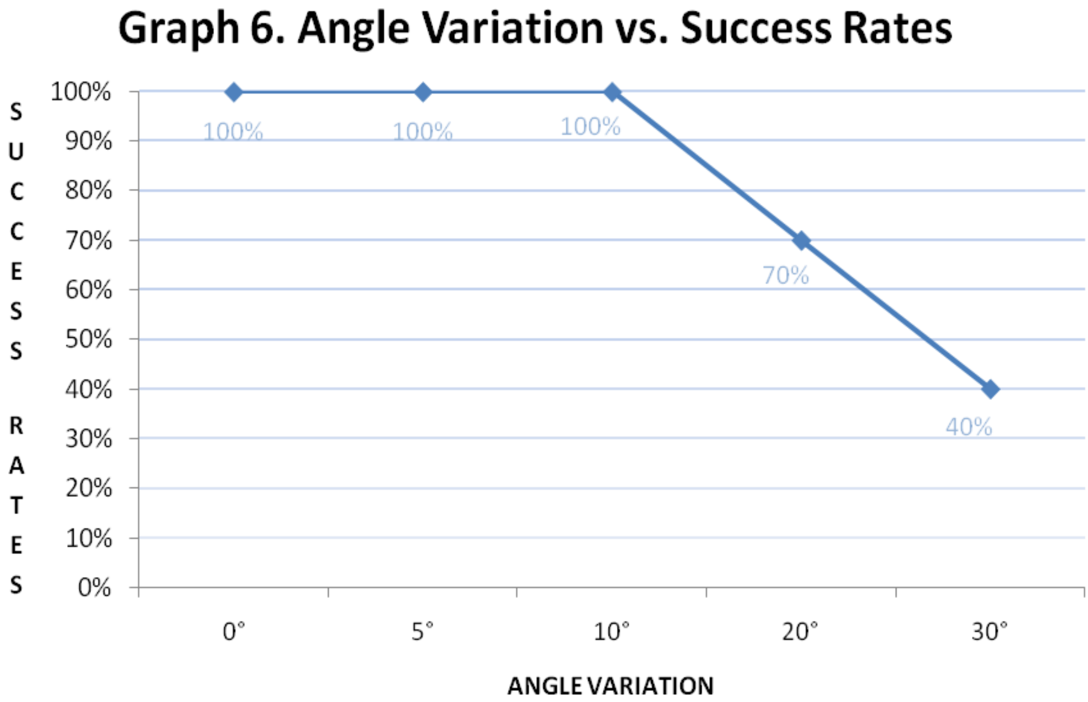
Graph 6-5. Scaling Variation Effects for the System of Z. Saifi and S. Sadhu

As for the third column of the chart, which states the ideal condition, all the results should be correct of course and we get what we expect from the software. For the next column the success rate lowers respect to the third but raises respect to the second one and this result of forty percent, is lower than the result of the BPN software. For the last column, the success rate drops down thirty percent affirming our idea that states increasing in success rate with shorter distance between camera and the ace up to a limit. Because, once again, the success rate increased from beginning to end (disregarding the ideal condition) until the 105% limit was reached. After this limit the rate starts to lower again. Thus, we can conclude commenting on the scaling variation graph by saying the tests were generally suitable and the final tests of angle variation are going to play the important role for comparison between this software and the BPN based one from now on.

The Effect of Yaw Angle Variation to the Face Recognition System by Z. Saifi&S. Sadhu					
	0°	5°	10°	20°	30°
1	1	1	1	1	32
2	2	2	2	5	5
3	3	3	3	2	2
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	15	15
7	7	7	7	7	24
8	8	8	8	8	8
9	9	9	9	9	4
10	10	10	10	10	10

Chart 6-8. The Effects of Yaw Angle Change to the System of Z. Saifi and S. Sadhu

On the previous chart there is no need for discussing the first column again since it shows the ideal condition where there is a hundred percent of success rate. But this success is again achieved for the five and ten degrees of angle changes. This shows that for small angle changes the system is really successful and never makes a mistake for our database. When we increase the angle more we expect some mistakes and it happens so the success rate degrades to seventy percent. This rate can also be accepted as a success if we consider it is not easy for a program to recognize faces with wide angles.



Graph 6-6. Angle Variation Effects for the System of Z. Saifi and S. Sadhu

As it can be seen from the graph the success rate lessens and gets a value of forty percent. For the graph, it can be noted that there is a good distribution of success rates and angle changes; therefore, we are not seeing any incoherence considering the fact that success reduces as wider angles are used. Until the twenty degrees of angle change, there is a straight line that shows the success that is over the top. Finally what we can say for this software is that it is not back from the BPN based face recognition system in angle variation on account of good results.

7. Final Comparison of the Best Algorithms

In this section, the final comparison is being made. In the beginning of our work, the five working interesting programs with different approaches were selected among the ones that attract attention and worth to analyze. Then three of them were finally thought for the extra tests. Among those three, the Fisherface Face Recognition system which was designed by A. H. Omidvarnia was also eliminated due to full failure for a forty imaged database. For this software, also other tests were made with ten imaged training database in order to simplify the process and try to understand some aspects of the program. However, again in those tests it was clear that this software is obviously worse than the other two selected. Because for the tests have been made with different angles, the results showed that the Fisherface approach cannot reach the rates that are achieved by other two. Thus, the final comparison is supposed be done between Back-propagation Neural Network (BPN) based Face Recognition Program and Face Recognition System by Z. Saifi and S. Sadhu. This time we can perform a better analysis by adding new sharp values to our tables. These values will be in between of the previous ones. Here the idea is that presenting a better detailed analysis by doing the tests also for minor diversity differences.

The Effect of Yaw Angle Variation to the BPN System									
	0°	2.5°	5°	7.5°	10°	15°	20°	25°	30°
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	12	5	5	5	5
3	3	3	3	3	3	3	3	2	2
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	15	9	15	15
7	7	7	7	7	7	7	7	24	24
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	4	4
10	10	10	10	10	10	10	10	10	10

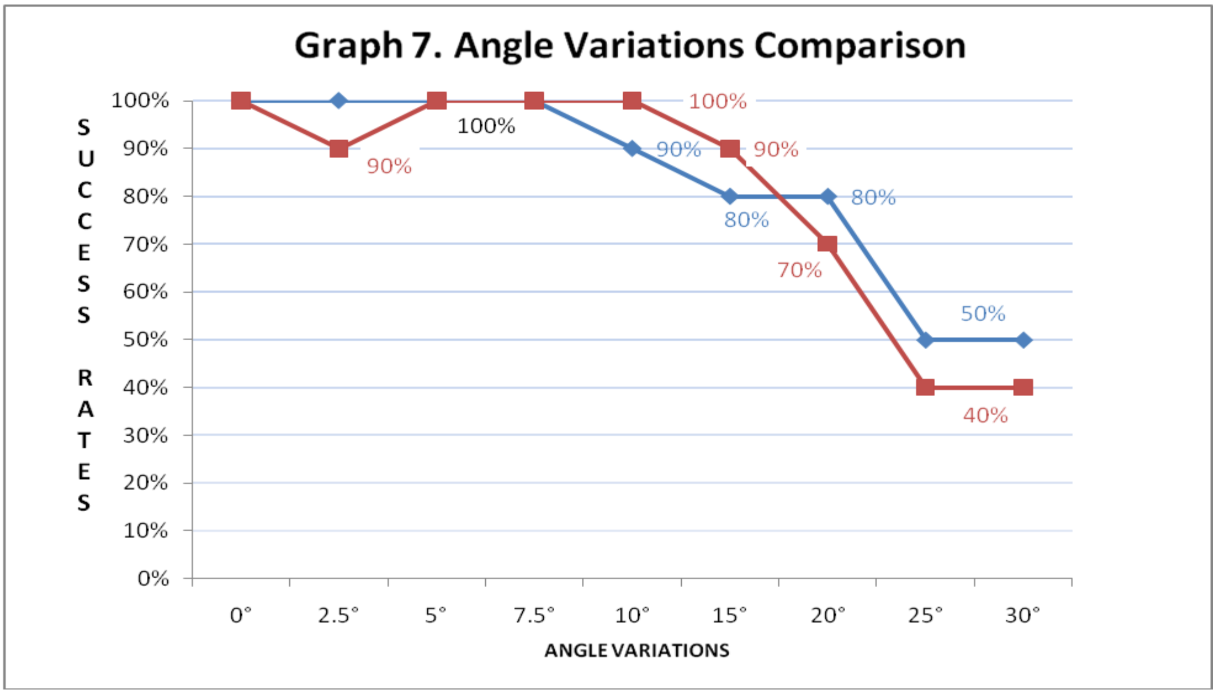
Chart 7-1. The Results for Yaw Angle Change for the BPN System

We can start comparing them with the performance in angle variations. In angle variations for zero and five degrees, they have same result. There is an interesting result for Z. Saifi's software which is ninety percent for 2.5 degrees. However, the results for the zero degree and five degrees have both a hundred percent success rates. It should be noted that this is because the number of test have been performed. This means if we performed many more tests, for example a hundred, then the results for three of them would have been vicinity of a hundred but they would be less than a hundred probably and also in that case the result for 2.5 degrees would be greater than ninety percent. Thus, since the software caught the a hundred percent rate again for the five degrees, we can ignore that mistake and think that as if a hundred. Then another difference appears when the angle is ten degree. This time for BPN, there is one failure so the BPN approach falls behind at first because for other software there is still a hundred percent success rate. However, this condition changes back when the angle is twenty degree.

The Effect of Yaw Angle Variation to the Face Recognition System by Z. Saifi&S. Sadhu									
	0°	2.5°	5°	7.5°	10°	15°	20°	25°	30°
1	1	1	1	1	1	1	1	32	32
2	2	2	2	2	2	2	5	5	5
3	3	3	3	3	3	3	2	2	2
4	4	4	4	4	4	4	4	4	4
5	5	2	5	5	5	5	5	5	5
6	6	6	6	6	6	15	15	15	15
7	7	7	7	7	7	7	7	24	24
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	4	4
10	10	10	10	10	10	10	10	10	10

Chart 7-2. The Results for Yaw Angle Change for the System of Z. Saifi and S. Sadhu

Also on the following graph it can be observed that blue colored line, representing BPN approach, passes through the red graph line (which indicates the performance of the software designed by Z. Saifi and S. Sadhu) and stays over the red colored line. Thus this graph shows that BPN based software is slightly better than the other one with respect to angle variation.



Graph 7-1. Performance Comparison in Angle Variations for the Best Softwares

Before our last comment we should analyze the performances also for the scaling changes. If the result is the same like the angle variation tests, there is nothing left us to think or consider. Thus, it is useful to use another graph to see the results together. Again in this case we can add other values to our tables which will show us better graphs due to new corresponding results.

The Effect of Scaling of the Image Size to BPN System									
	90%	92.5%	95%	97.5%	100%	102.5%	105%	107.5%	110%
1	11	11	20	1	1	32	32	13	13
2	20	1	1	2	2	5	5	5	5
3	20	20	1	30	3	33	33	22	22
4	8	6	6	4	4	4	4	4	4
5	20	34	34	5	5	5	5	5	5
6	8	6	6	6	6	15	9	15	15
7	15	11	11	7	7	24	24	24	24
8	8	8	8	8	8	8	8	8	8
9	8	6	6	9	9	9	9	35	35
10	11	11	7	10	10	10	10	13	13

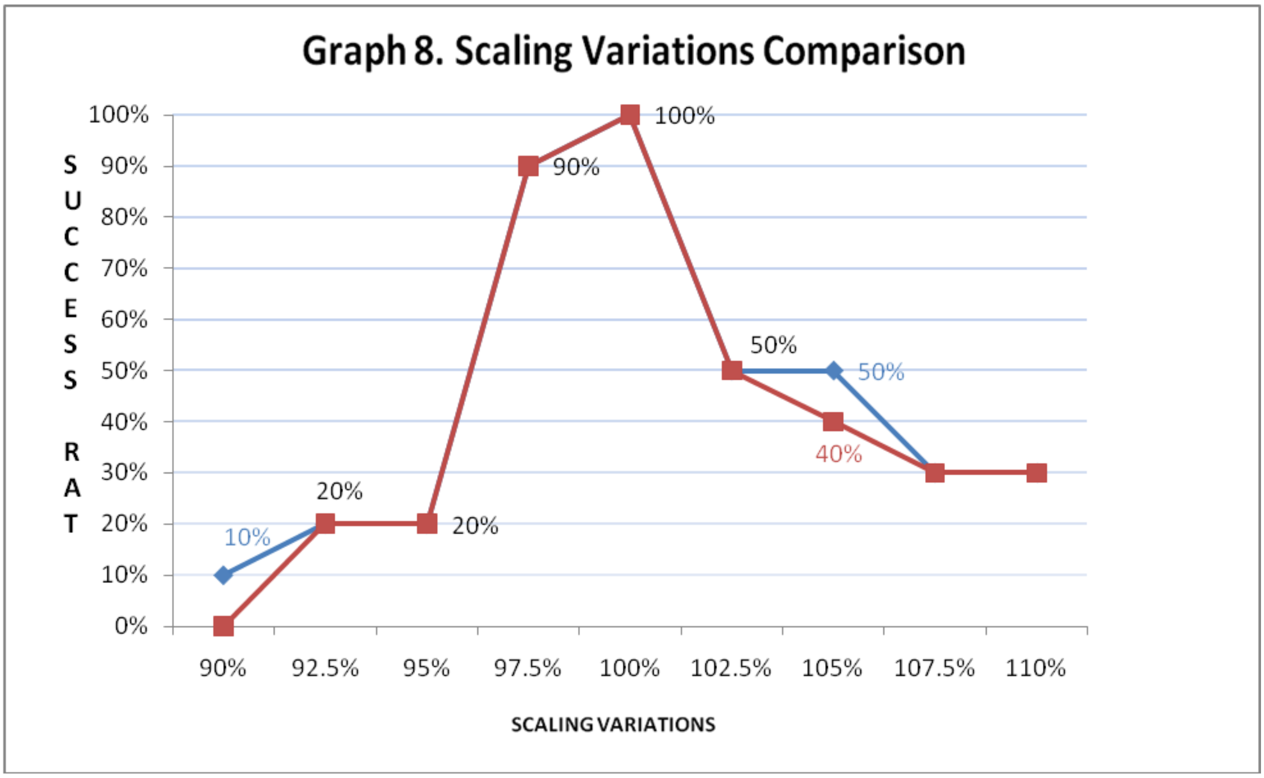
Chart 7-3. The Results for Scaling Change for the BPN System

As it can be seen from the new tables, there are new values added as 92.5%, 107.5% in order to understand the programs' response in shorter value range. With use of next table and the previous one, a new better graph can be drawn showing the programs' success rates under the scaling variations.

The Effect of Scaling to the Face Recognition System by Z. Saifi&S. Sadhu									
	90%	92.5%	95%	97.5%	100%	102.5%	105%	107.5%	110%
1	11	11	20	1	1	32	32	13	13
2	20	1	32	2	2	2	2	5	5
3	20	1	30	30	3	33	22	22	22
4	NMF	6	6	4	4	4	4	4	4
5	20	1	32	5	5	2	5	5	5
6	NMF	6	6	6	6	15	15	15	15
7	15	11	11	7	7	24	24	24	24
8	NMF	8	8	8	8	8	8	8	8
9	NMF	6	6	9	9	9	15	35	35
10	11	11	7	10	10	10	13	13	13

Chart 7-4. The Results for Scaling Change for the System of Z. Saifi and S. Sadhu

On following graph the scaling variation performances are drawn and the blue color again represents the performance of the BPN based approach and the red color indicates the Z. Saifi's software which uses Eigenface approach for face recognition.



Graph 7-2. Performance Comparison in Scaling Variations for the Best Softwares

This time the lines do not start from the same value. For our first part of the test, BPN software starts better but for second and third part, the Z. Saifi's software catches the same amounts. But it is obvious that red line does not pass over the blue one, and moreover according to results for 105% of scaling change, it falls behind the blue line once again. They come together at the next tests but overall results show that BPN based approach is again slightly better. This confirms what we had found under the other different conditions. Hence, in almost all area, the BPN based face recognition software is more successful and better than the others. Z. Saifi and S. Sadhu's Eigenface based face recognition method comes as the second.

As to some important general results, from the result obtained during the last tests we can see that; as the distance between camera and face increases the success rate dramatically decreases, but when the distance decreases, so when the face gets bigger, the success rate does not decrease too dramatically so we do not get a sharp slope in the graph like in the previous case. Finally for the last angle variation results, it can be seen that the success rate starts to change more sharply towards downward after twenty percent of angle change. Thus it can be said that greater angle changes start to cause huge differences between success rates after some limit which is twenty for these softwares.

8. Concluding Remarks

The aim of this project has been a research of personal identification through face recognition. For this aim the different face recognition softwares have been analyzed for understanding the properties and success rates of each of them under the different conditions while image acquisition or varied sources. The changes of the sources were operated by the help of the software which can create 3D human face models named "Facegen Modeller". Before explaining the program, in this work, the basic concepts were well studied to understand the basic knowledge about face recognition. Moreover, the history of this face recognition subject is also so significant since the developments can be only understood in this way and also to have an idea about the application fields of face recognition hence these subjects have been thought to be present in this whole work. The calculation of Eigenfaces also was particularly important for our case when we have to deal with Eigenface method but also other methods that can be related with this. Since the comparison of algorithms is also the another result expected from this work, the approaches were classified in a general way as under the titles of direct correlation, Fisherface, and Eigenface methods and later the general fundamental knowledge were added for each of them and then the part was completed by information about how to calculate the Eigenface.

The first thing done by this "Facegen Modeller" was the creation of a totally new database which can be used for later operations and tests. By this way the gathered database could have been changed when needed, for instance, when it is needed to be tried in order to get if it is more sensitive to expression changes compared to other variations. It means that the ease of using of the same database was also provided by this software and by this method it was not necessary to create new face images all the time from the very beginning. Instead the same initially created and saved human face models were used by changing just the other properties. Otherwise, it would be so difficult to create face images over again for each test and the reliability would be so low since it is sort of farfetched to expect the same faces would be created because in that case the face creation process would start once more and the complexity of human face would prevent to reach same details on the face. Thus, it can be said that the new face would never be the same again. Therefore, the advantages of altering the each parameter separately to analyze the effect of them are well utilized by using the same database from beginning to end. As to most crucial and the last big part of this work which is testing the softwares there are two different conditions were first considered. These are the ideal conditions and the real conditions which state that more realistic conditions and can be called varied conditions or with other words, the uncertainty sources such as scaling, expression, viewpoint, and illumination that are provided by the usage of Facegen Modeller software. Whereas in the ideal conditions the same images of the each different people which means for example each person's same poses so same images (without any change in scaling, viewpoint etc.) were used exactly in both training and test sets and the ideal conditions were preserved with this way and so that the tests were made under the ideal conditions as the name implies.

In this work the carefully selected five programs were tested and the results in ideal conditions were generally all good. Among these five softwares, the first one, Drexel University's software, was kept separated than the others due to its different purpose of usage which is worth to examine but not so related with our final goal. Among the rest four softwares it was seen that the BPN based software has a disadvantage as it is time consuming for each and every test. However, the results for both realistic and ideal cases were really good. Finally it was found that its weakest point was illumination but the best point was the success in the change of angle and expression. These results were nearly reached by also the last software tested which uses Eigenface approach and those two softwares and also the Fisherface software from A.H.Omidvarnia, which has dramatically lower error rate than its

Eigenface version, were selected for later tests. From that point on these softwares were going to be used for further two tests which were going to be conducted by same database again. Therefore the tests were ended up with these softwares under the effect of scaling change and the angle variations. After having the results, it was seen that the best two softwares (BPN and the System by Z. Saifi and S. Sadhu) made their difference. In the end we can say that those two softwares were analyzed in order to understand their performances for different conditions also for more strict conditions and BPN based was chosen as the best one, moreover some important results for both softwares were used for commenting on some outstanding results concern with face recognition.

Acknowledgements

I would like to thank all the people who helped and inspired me during my thesis study. I am sincerely thankful to my supervisor, Prof. Emanuele Zappa for guidance and encouragement from the beginning to the final level. This support allowed me to develop a better understanding on "Face Recognition" and to work better and detailed on my subject. I am also grateful for his advices and everything he has done in order to increase my willingness to study hard.

I would like to thank my friend Funda Cevik as well for her efforts to read my work over to make a comment that led me present a better thesis.

Beyond everything I owe my deepest depth of acknowledgement to my dear parents for an endless source of love, patience, encouragement, and support.

References

[1]J. Barron, N. Pears, Feature, “Feature Localization in Three-Dimensional Faces”, Project, MEng. in Computer Systems and Software Engineering Department of Computer Science, University of York, May 2008

[2]T. Heseltine, N. Pears, J. Austin and Z. Chen, “Face Recognition: A Comparison of Appearance-Based Approaches”, Department of Computer Science, University of York

[3]P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces”: Recognition Using Class Specific Linear Projection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, July 1997

[4]I. Alphonso, “Heteroscedastic Discriminant Analysis”, Critical Review Paper, January 2000

[5]M. Turk, A. Pentland, “Eigenfaces for Recognition”, Journal of Cognitive Neuroscience Volume 3, No. 1, Massachusetts Institute of Technology, 1991

[6]I. Atalay, “Face Recognition Using Eigenfaces”, Msc. Thesis, Drexel University, 1996

[7] S. Anam, M. S. Islam, M. A. Kashem, M. N. Islam, M. R. Islam, and M. S. Islam, “Face Recognition Using Genetic Algorithm and Back Propagation Neural Network”, Proceeding of the International MultiConference of Engineers and Computer Scientists 2009 Volume 1, IMECS 2009, March, 2009, Hong Kong

[8] Eigenface tutorial, Drexel University, Retrieved from http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm

[9]A. Ozdemir, “Face Recognition of Frontal Human Images by Using Wavelet Transform”, Msc. Thesis, Department of Science, University of Kahramanmaras Sutcu Imam

[10]L. Sirovich, “Image Analysis”, Tutorial Part 1, June 2001

[11]P. Latha, Dr. L. Ganesan, and Dr. S. Annadurai, “Face Recognition using Neural Networks”, An International Journal (SPIJ), Volume 3, Issue 5