Bilkent University

Department of Computer Engineering

Senior Project

Who do you resemble?

Final Report

Group Members

Merve Soner
Merve Yurdakul
R. Baturalp Torun
Sedef Özlen

Supervisor: Pinar Duygulu Şahin

Jury Members: Selim Aksoy, H. Altay Güvenir

May 20, 2008

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Projects course CS492.

Table of Contents

Table of Figures

1 INTRODUCTION

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. There are many usage areas of face recognition which are mainly security systems, robotics and personal areas such as tagging photos. Although, there are many algorithms, none of them guarantees good face recognition. Therefore, face recognition topic is a hot topic which is yet to improve.

Popular recognition algorithms include eigenface, fisherface, the Hidden Markov model, and the neuronal motivated dynamic link matching [1]. A newly emerging trend, claimed to achieve previously unseen accuracies, is three-dimensional face recognition. Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. Our aim is to find the most accurate working method among these and improve it.

We pursued a different approach than the existing ones. We discovered that among these methods Lowe's Sift method for object recognition gives the best results although it is for object recognition. We modified this algorithm for face recognition. We tested our algorithm by using an application called "Face Finder". In this application, given photos are compared among a large data set which consists of celebrities, and the celebrities that resemble the given photo most are returned. This operation is done by finding the interest points of the faces and matching these points. Through this report we will give the implementation details of the implemented algorithm, as well as the results and the efficiency of the algorithm with charts, pictures, etc.

2 MOTIVATION

2.1 Purpose of the System:

Human brain is very developed in terms of face recognition. Face recognition ability starts to develop from a few weeks after birth. We meet hundreds of people during our lives and we keep their faces in our brain. When we see, these people again, our brain can recognize their faces. Even if we see a person for once we can remember this face when we again see it or even if the out looking of the people we know change, our brain can still recognize these faces.

However, face recognition problem is a difficult task for computers and it is an unsolved problem of artificial intelligence and computer vision area. Computers are not capable of recognizing faces among hundreds of face dataset. However the face recognition problem is a large area and we narrow it down by using face matching for face recognition.

In our project, we try to find the similarities between face pairs to find the face that resembles the most to the given image.

2.2 Importance of Face Recognition:

Face recognition can be used in a wide area from security, robotics to personal photo/video organization or context based video indexing. The diversity of the usage area and the importance of the task make face recognition important for the technologic world. It can be used for criminal recognition or airport security as well as funny applications like image searching, automatic image tagging, etc.

2.3 Problems of Current Algorithms:

Current face recognition algorithms work with controlled face images in terms of pose, illumination, etc. A learning mechanism is performed with small and controlled dataset to construct the training set. Then the face recognition process is performed with this small training set. In order to enlarge the dataset, a large amount of manual work is required and as the manual work increases, the algorithm becomes more error prone.

With rapidly developing applications, there are many uncontrolled face images in the Internet. They are not suitable for the current face recognition algorithms and there is not any efficient way to perform face recognition in these images.

Thus, we come up with FaceFinder Library which is a flexible open source library that can be used in many applications.

3 OUR APPROACH

Each person has distinct facial features that do not change with age, makeup, hair style, etc and these distinct features can be used for face matching. The distinct feature concept is commonly used for object recognition in previous studies and it gives a good performance. By combining these two ideas, we decided to adapt distinct feature concept for face recognition.

In this project, we use Lowe's Sift keypoint detector (see Appendix A) to extract the interest points of the face images. This method takes two image files as input and extracts the interest points of these image files. Then, it performs a match point detection (see Appendix A) for the extracted interest points. Figure.1 shows the working process of this method.

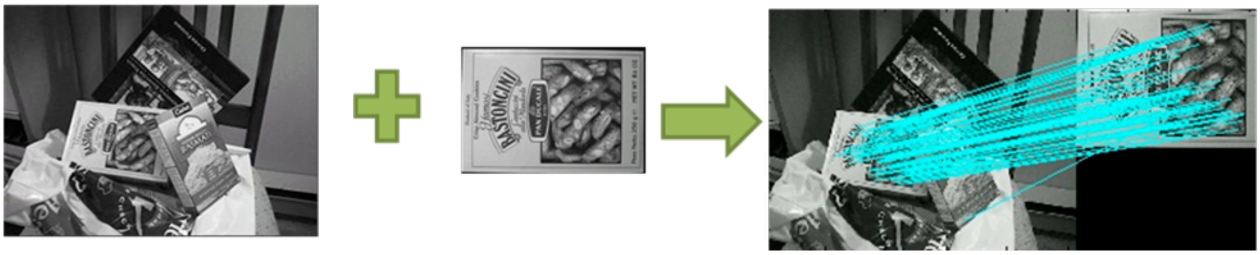


Figure 3-1 – Lowe's SIFT Keypoint Detector.

As it is shown in Figure.3-1, Lowe's keypoint detector performs very well for object recognition. It can find approximately all matching interest points of the given images with little errors. The reason for such a good performance is that the deformation of the objects is not so complicated. Position and illumination of the objects may change during the object recognition process and such changes don't affect the matching process seriously.

When we try to directly use this method for face recognition problems, some problems have occurred because of the deformation of the face images. Changes in face images are not as simple as in objects. When a person changes his hair style, wears glasses or makeup, has his hair or moustache cut, etc, his outlook changes dramatically and Lowe's SIFT method gives poor results for interest point matching.

Results of the Lowe's SIFT keypoint extractor's were not satisfying and they do not meet our requirements. Therefore, some additional steps are used while developing the project. These steps help to improve robustness of the project. There are 5 steps in total and they are explained in details in the following sections.

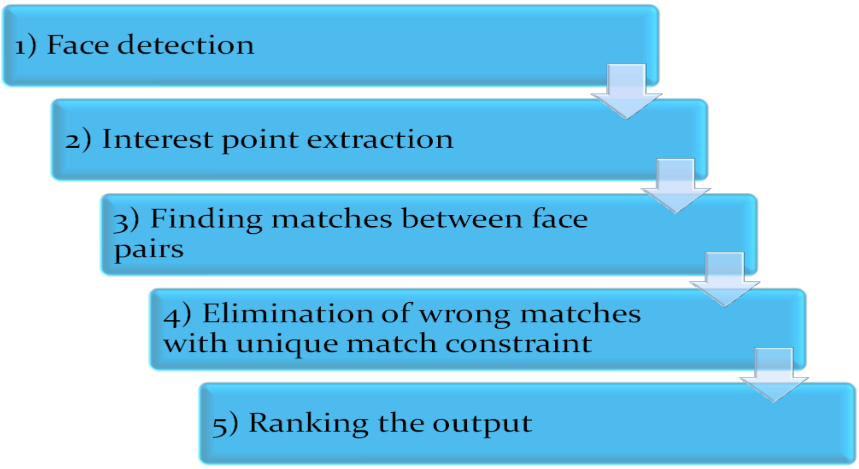

1) Face detection
2) Interest point extraction
3) Finding matches between face pairs
4) Elimination of wrong matches with unique match constraint
5) Ranking the output

Figure 3-2 – Five steps of face recognition in our approach

3.1 Face Detection

Face detection is the main requirement for face recognition problems. All the faces need to be detected in order to start face recognition. Therefore all images are processed by a face detector. OpenCV, which is an open source computer vision library, is used for the project. It is a popular and authentic library developed by Intel. It has various packages for different tasks in computer vision but face detection module is used for the senior project. The system detects the face in given image and returns coordinates of upper left and bottom right corners of the face. Therefore face area is extracted from the given image. OpenCV can detect multiple faces in a given image. However it is disabled since the project does not require multiple face matching from the given image. After face is extracted from image, given image file is resized proportional to the size of face in given image. It needs to be resized because Lowe's keypoint extractor may produce many interest points that are totally useless. Therefore image resizing helps to eliminate useless interest points at first step and reduces the computation complexity. On the other hand, small faces must be magnified in order not to lose important interest points. After resizing process, all faces are converted into grayscale since RGB colors are not important for feature extraction. Project only deals with intensity ratio of color tones which 256 (grayscale) tones are enough. An example of a sample output of face detector code is shown on figure 3.1-1.

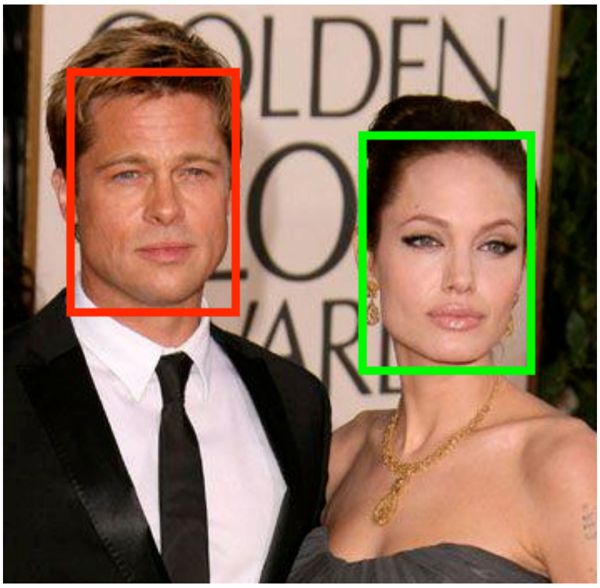


Figure 3.1-1 – Face Detection

3.2 Interest Point Extraction

All face images are processed by Lowe's keypoint extractor and each image is represented by a keypoint file. Therefore Lowe's keypoint extractor runs for once and extracts interest points for each face image. Extracted interest points for an image pair are shown in Figure.3.2-1.

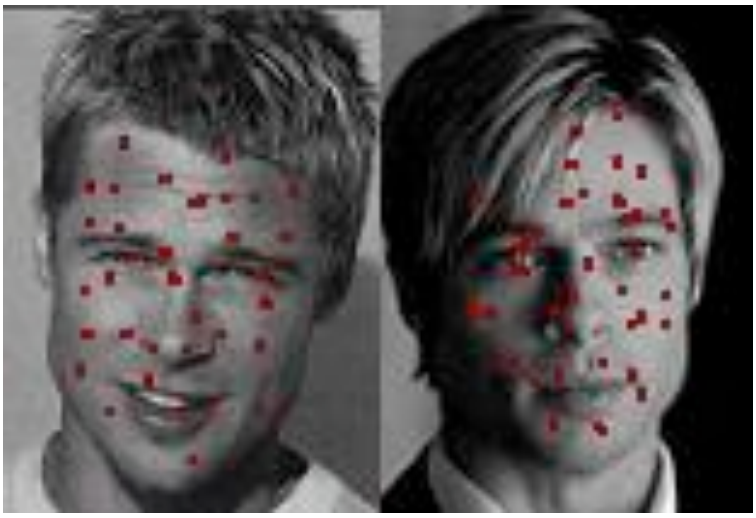


Figure 3.2-1 – Interest Point Extraction

3.3 Finding Matches between Face Pairs

Extracted interest points are needed to be matched between face pairs however Lowe's method only works fine on object recognition. Therefore it produces many useless and poor matches between irrelevant interest points on faces. It can be seen from Figure.3.2-1, an interest point in hair matches to mouth and orientations of the lines are not horizontal. As matching algorithm has already designed for object recognition and objects can be oriented differently in various images. However, in an ideal face matching system, lines should be horizontal because all faces are horizontally oriented well and looking ahead. Therefore some poor match elimination methods and constraints should be used in order to reduce number of poor matches. Irrelevant matches divert average mean values and it causes to match wrong face pairs.

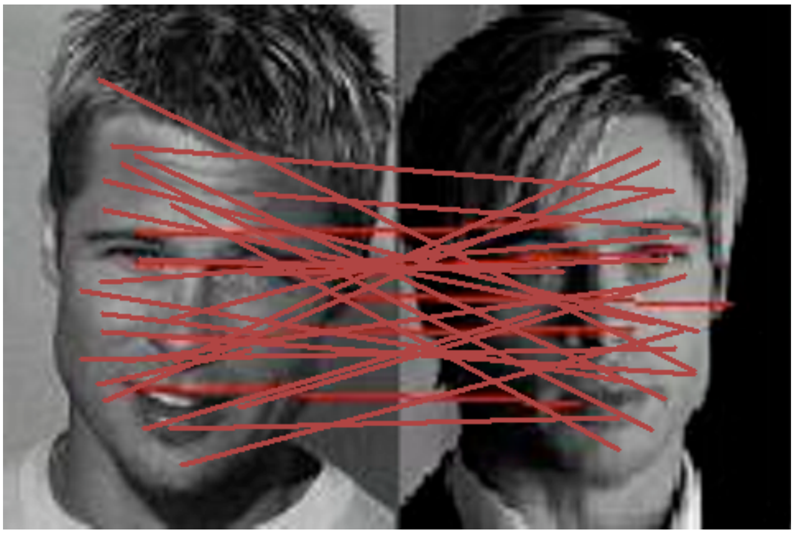


Figure 3.3-1 – Interest Point Matching

3.4 Elimination of Wrong Matches with Unique Match Constraint

Unique match constraint is used to reduce number of wrong matches. All the matches must be one-to-one and onto in a face pair. Hence one-way assignments are eliminated between interest points. Only two-way assignments are acceptable as seen in Figure.3.4-1.

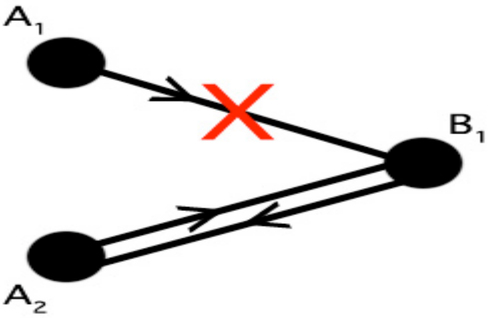


Figure 3.4-1 – Elimination of One-way Assignments

On the other hand, if two points are matched with one point such as A1 & A2 then one of them should be eliminated as shown in Figure.3.4-2. In this case, values that are calculated by Euclidean distances are used to eliminate them. Smaller distances are preferred for better matching between face pairs.

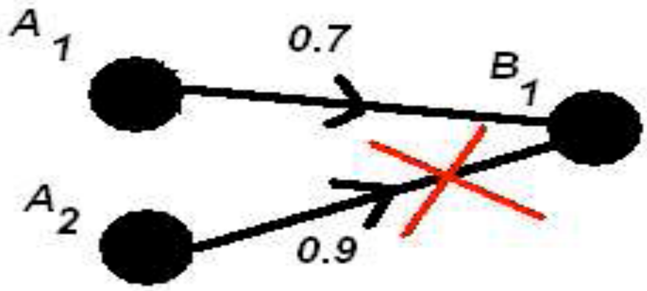


Figure 3.4-2 – Elimination of Multiple Assignments

Consequently, unique match constraint definitely improves matches between face pairs and eliminates poor matches. As it can be seen from Figure.3d-3, remaining matches are almost horizontal and reasonable matches such as eye-to-eye, nose-to-nose, lip-to-lip etc. After poor matches have been eliminated, average mean for each face is calculated by using averages of remaining matches.

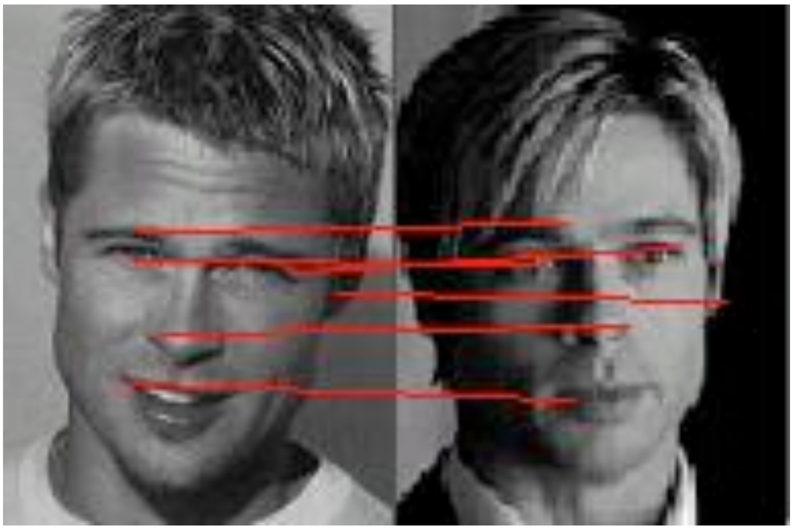


Figure 3.4-3 – Elimination of Wrong Matches

As we stated in previous reports we have tried geometrical constraint beside unique match constraint. Nevertheless the results varied in different inputs. We got good results for some inputs but it was not the same for all inputs. As we could not get any stable value for geometrical constraint we decided not to use it.

3.5 Ranking the Output

First 4 steps are done for all images in the database and computation should be done for given query. After all average values are calculated for each face pair then they should be ordered properly in order to present a reasonable result set. Average mean values are sorted in ascending order and minimum value is best match for the face query as shown in Figrue.3.5-1. User is usually informed by best match and next 10 matches.

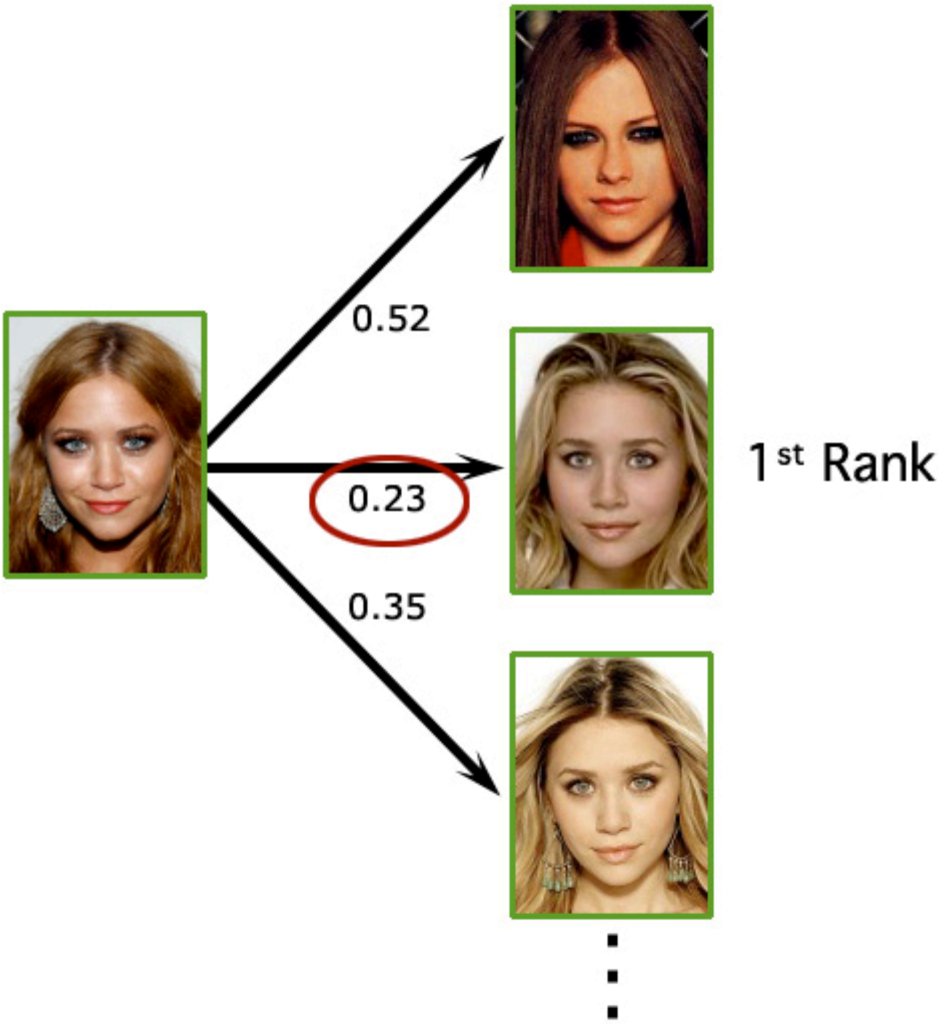


Figure 3.5-1 – Ranking the Output

4 APPLICATION DETAILS

As it is seen from the figure 4.1 -1 there is 6 steps in the application. Appendix B – User's Manual gives detailed information about each step of the application. Whereas, here it is given the details of the implementation of the application.

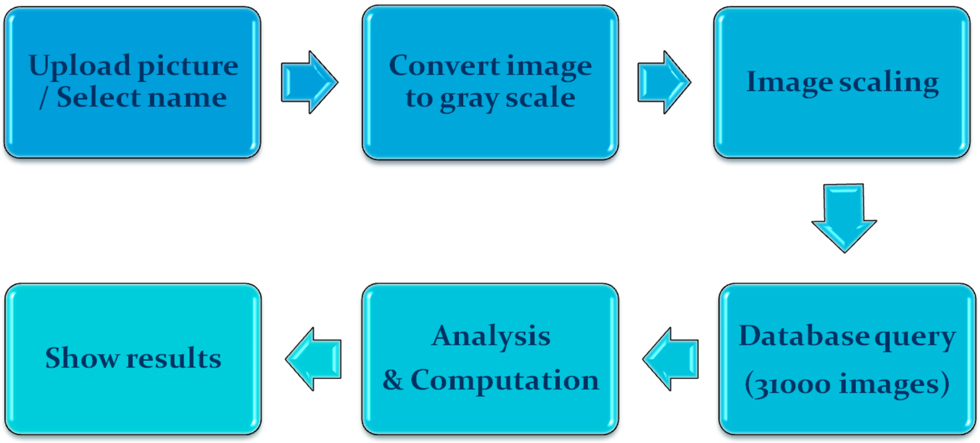

Upload picture / Select name
Convert image to gray scale
Image scaling
Database query (31000 images)
Analysis & Computation
Show results

Figure 4-1 – Six steps of application

Implemented Classes:

- Match.cpp: Performs the main computation of the system. It matches the given image with the other imaged in the dataset.

Functions:

Image ReadKeyFile(const string& inputfile): Reads the key file of a specific image and initializes the properties of Image object.

void Tokenize(const string& str,vector<string>& tokens,const string& delimiters): Tokenize the given input file and initializes the size of the image.

double FindMatches(Image im1, vector <KeyPoint> keys1, Image im2, vector <KeyPoint> keys2, string result_path): It takes two images and their keypoints to find matches between the images. If a result path is given, it creates an image that shows the matchings between images.

KeyPoint CheckForMatch(KeyPoint key, vector <KeyPoint> klist): Finds the matching of a given keypoint among a set of keypoints.

int DistSquared(KeyPoint k1, KeyPoint k2): Calculates the Euclidean distance between two keypoints.

int **AllocMatrix(int rows, int cols): Allocates an integer matrix.

float **AllocMatrix2(int rows, int cols): Allocates a float matrix.

Drawable_Image *ReadPGMFile(const char *filename): Reads the given image file pixel by pixel.

void WritePGM(FILE *fp, Drawable_Image *image): Creates a pgm file.

void DrawLine(Drawable_Image* image, int r1, int c1, int r2, int c2): Draw line between the given coordinates.

Drawable_Image* CombineImagesHorizantally(Drawable_Image* im1, Drawable_Image* im2): Combines two image horizontally.

bool compareMean(Image x, Image y): Compare the means of two images.

- Image.cpp: Represents the image files with their properties like width, height, keypoints, etc. The functions of this class are trivial functions like get, set functions and they are not explained in details.

- Keypoint.cpp: Represents the keypoints of an image and holds the properties of a specific keypoint. The functions of this class are trivial functions like get, set functions and they are not explained in details.

- Drawable_Image.cpp: Represents the image files as pixels in order to draw their matching. The functions of this class are trivial functions like get, set functions and they are not explained in details.

- File_Reader.cpp: Initializes the compare set that will be used to find similar of a given face image. The functions of this class are trivial functions like get, set functions and they are not explained in details.

5 TESTS & RESULTS

In order to test our system we have used several test cases to determine how well this new approach to the face recognition is working for different inputs and for different data sets. Before we have tested our system by using the developed application we have also tested the method used at inner steps such as we compared the outputs when face detection is used and not used or when unique constraint is used and not used or when geometric constraint is used or not used. In figures 5-1, 5-2, and 5-3 we will see the results of the face recognition program when face detection, unique match constraint or geometrical constraint is used. To see the performance in these three inner cases we have given several images of a person and formed a chart that shows in which place it finds the same person's images out of 200 images. We have given '+' symbol for representing the images of the person that is being tested and for the other images we have given '.' symbol.

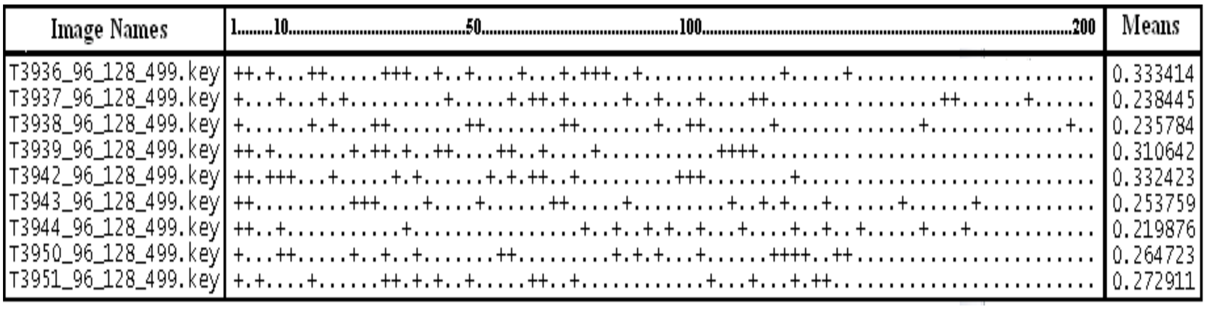


Image Names	1........10..50...100...200	Means
T3936_96_128_499.key	++.+...++.....+++..+..+....+...+.+++..+.............+.....+.......................	0.333414
T3937_96_128_499.key	+...+...+.+.........+.....+.++.+.....+..+...+....++................++......+......	0.238445
T3938_96_128_499.key	+......+.+...++.......++.......++.......+..++......+..............+.............+..	0.235784
T3939_96_128_499.key	++.+.......+.++.+..++....++..+....+...........++++................................	0.310642
T3942_96_128_499.key	++.+++...+.....+.+......+.+.++..+.........+++........+............................	0.332423
T3943_96_128_499.key	++.........+++....+....+......++.....+.........+..+.+...+......+......+...........	0.253759
T3944_96_128_499.key	++..+...........+................+..+..+.+..+...+....+..+..+.....+...+............	0.219876
T3950_96_128_499.key	+...++.....+..+..+.......++.........+.+.+...+......++++..++.......................	0.264723
T3951_96_128_499.key	+.+....+......++.+.+..+.....++..+.............+...+...+.++........................	0.272911

Figure 5-1 – Output after face detection is applied with unique match constraint

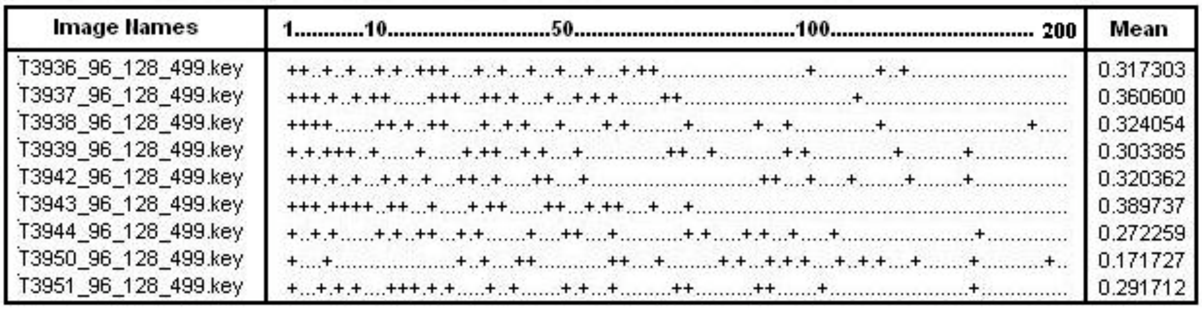


Image Names	1............10..............................50..100.................................... 200	Mean
T3936_96_128_499.key	++..+..+...+.+..+++....+..+...+...+...+....+.++.........................+.........+..+...........................	0.317303
T3937_96_128_499.key	+++.+..+.++......+++...++.+....+...+.+.+.......++............................+...................................	0.360600
T3938_96_128_499.key	++++.......++.+..++.....+..+.+....+.......+.+.........+.........+...+.............+.......................+.....	0.324054
T3939_96_128_499.key	+.+.+++..+......+......+.++...+.+....+.............++...+..........+.+.............+.........+................	0.303385
T3942_96_128_499.key	+++.+..+...+.+..+....++..+.....++....+...........................++....+.....+........+........+................	0.320362
T3943_96_128_499.key	+++.++++..++...+.....+.++......++...+.++....+....+..	0.389737
T3944_96_128_499.key	+..+.+......+.+..++...+.+.......+....++....+...........+.+.....+.+...+.....+.......................+.............	0.272259
T3950_96_128_499.key	+....+.....................+..+....++............++....+.........+.+...+.+.+....+..+.+....+........+..........+..	0.171727
T3951_96_128_499.key	+...+.+.+....+++.+.+.....+..+.......+.+...+.........++..........++.......+.......................+.............	0.291712

Figure 5-2 – Output after unique match constraint is applied

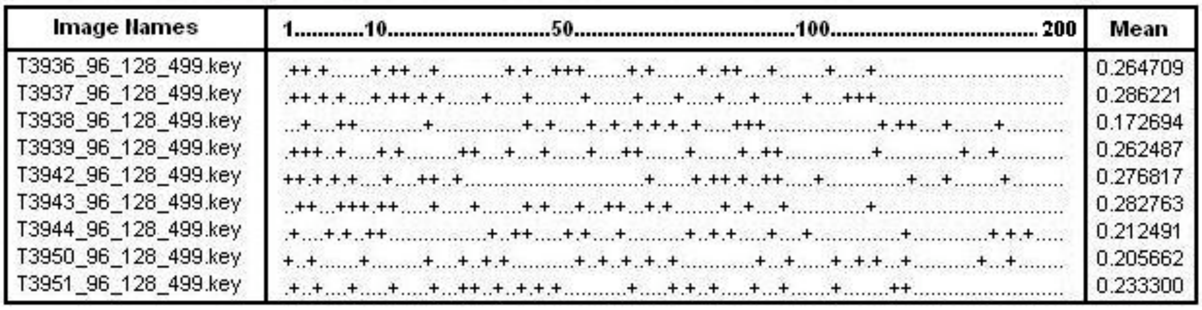


Image Names	1............10..............................50..100.................................... 200	Mean
T3936_96_128_499.key	.++.+.......+.++...+...........+.+...+++.......+.+.......+..++....+........+.....+...............................	0.264709
T3937_96_128_499.key	.++.+.+....+.++.+.+......+.....+........+.......+.....+.....+....+.......+.....+++...............................	0.286221
T3938_96_128_499.key	...+.....++...........+...............+..+.....+..+..+.+.+..+.....+++..................+.++....+.......+..........	0.172694
T3939_96_128_499.key	.+++..+.....+.+.........++....+....+......+....++........+.......+..++...............+.............+...+...........	0.262487
T3942_96_128_499.key	++.+.+.+....+....++..+............................+......+.++.+..++.....+.............+....+........+........	0.276817
T3943_96_128_499.key	..++...+++.++.....+......+.......+.+....+...++...+.+........+..+....+............+...............................	0.282763
T3944_96_128_499.key	.+....+.+..++.................+..++.....+.+....+..........+..+.+......+....+...............+.............+.+.+.....	0.212491
T3950_96_128_499.key	+..+.......+.........+....+..+.+..........+..+..+..+..+.............+...+......+..+.+...+...........+...+.........	0.205662
T3951_96_128_499.key	.+..+....+.....+.....+...++..+..+.+.+..........+.....+.+..+.....+...+......+........++.........................	0.233300

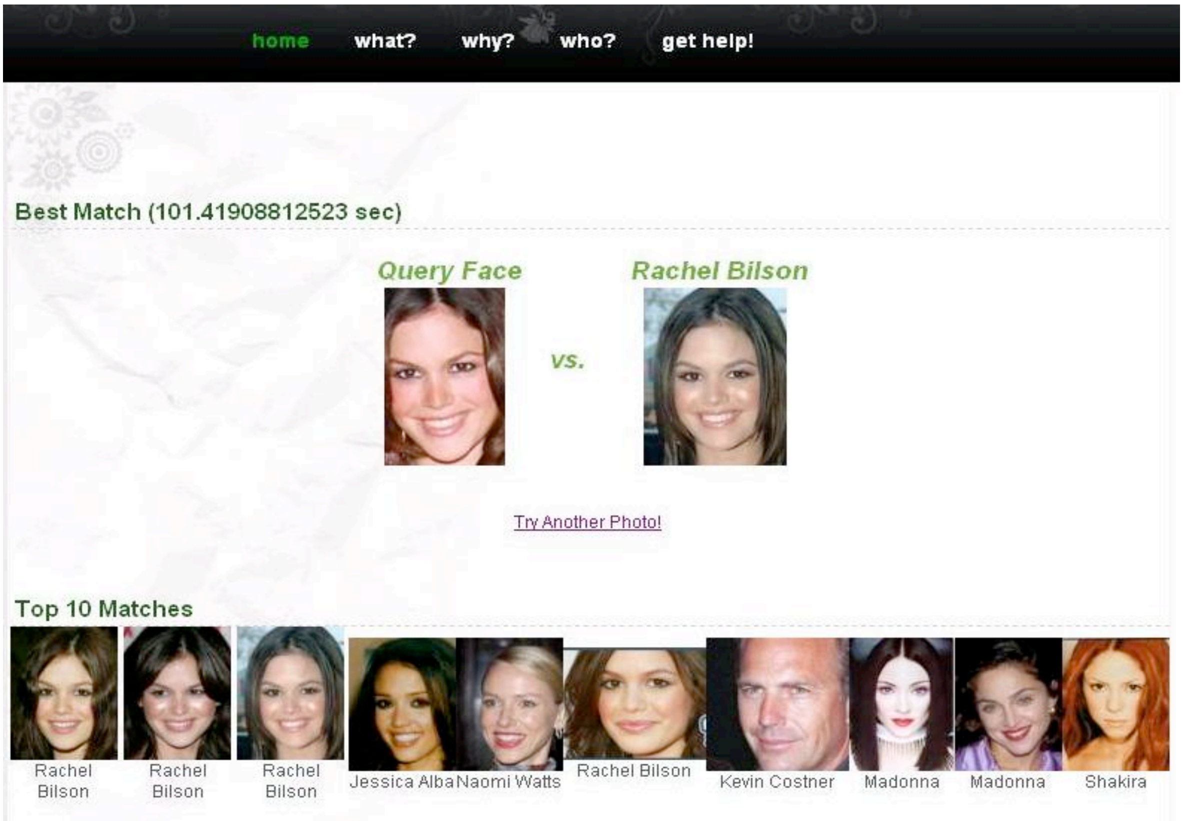
Figure 5-3 – Output after unique match constraint and geometrical constraint are applied

As we see from these figures we can see that face detection improves the results of the face recognition even if it is also important at some other aspects mentioned before. We also observe that geometrical constraint also improves the face recognition results for this input. Nevertheless it does not show the same improvement for all data sets therefore we eliminated this constraint as we mentioned before. In order to see more charts for different person images see Appendix C.

After we have tested the output for the inner steps of the development of the method, we focused on 4 different test cases to see how well the overall system works for different kinds of data sets. Firstly we have tested the current application with the images of the celebrities that are in the database but do not have the same images that are given as input, in the database. Secondly we have tested the age difference in the application in order to see whether it can find the same person at different ages. Thirdly we have given images with make-up and without make up to see if the system finds the same person at different conditions. Finally we have tried morph images to see the result and achive the performance.

5.1 Current Look

In this test case we have given current looks of celebrities that are exists in the database and try to see whether the program finds their different images in the first ranking and several other images in the top 10 listing. As we observe from the figure 5.1-1 when we gave a picture of Rachel Bilson to the program (where this image of Rachel Bilson does not exist in the data base but the database has 12 other pictures of her). As a result we see that the system finds the current look of Rachel Bilson in the first position as a best match. Additionally the system finds her other pictures that are in the database in the top 10 listing as first, second, third, forth and seventh among 15 images which is a good result. Same results are also observed for figure 5.1-2.



Best Match (101.41908812523 sec)

Query Face vs. Rachel Bilson

Top 10 Matches

Rachel Bilson, Rachel Bilson, Rachel Bilson, Jessica Alba, Naomi Watts, Rachel Bilson, Kevin Costner, Madonna, Madonna, Shakira

Figure 5.1-1 – Output for current look test

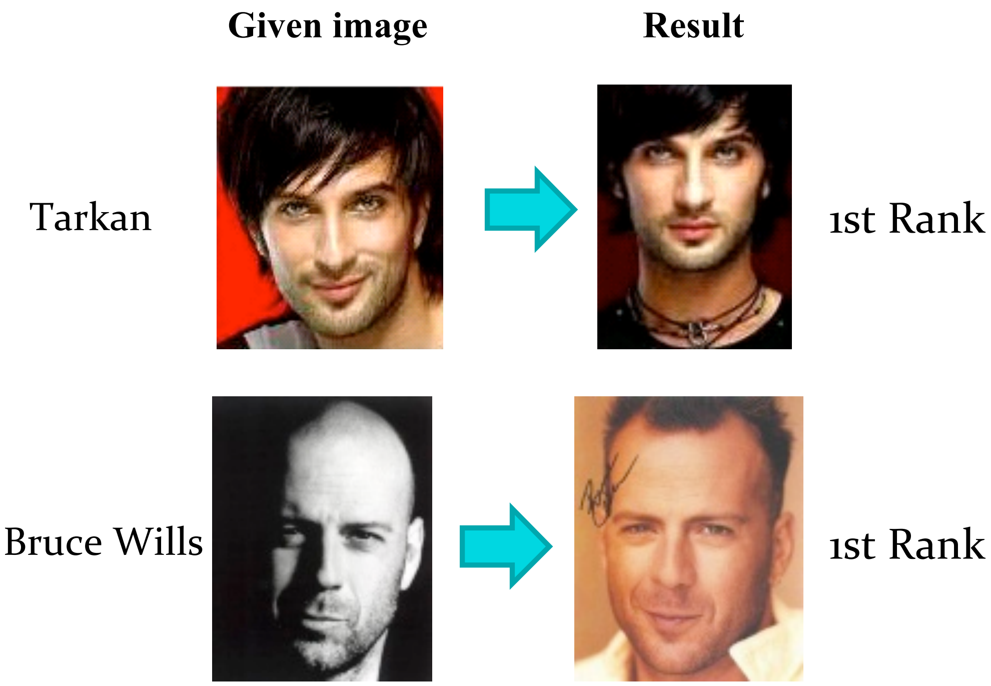


Figure 5.1-2 – Output for current look test

5.2 Age Difference

To test our application with a different set of inputs we have given images of celebrities at different ages in order to see if the system finds them correctly in a good ranking. As it is shown in figure 5.2-1 when an image of Elizabeth Hurley at the age of 17 we got her current look from the database in the first ranking as the best match. It is same for the input picture of George Clooney at the age of 23 the system returns an output of his current look image as first ranking. In the third case a graphical image of Nicole Kidman at an older age is given to the system and her current look is returned at the second rank.

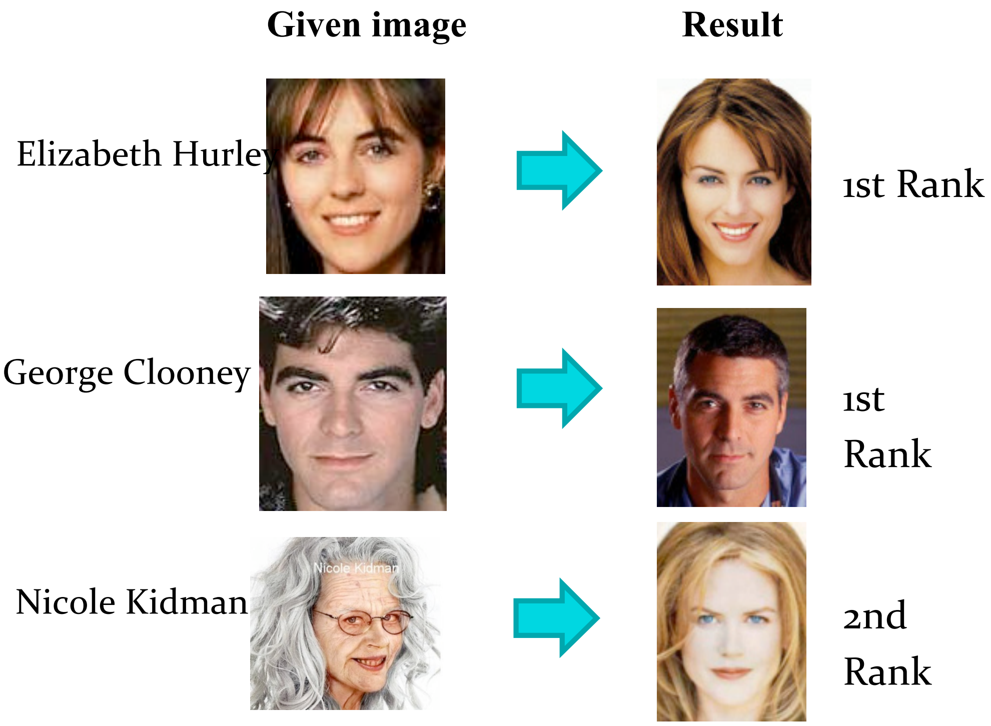


Figure 5.2-1 – Output for age difference test

5.3 Make-up Difference

In this test case we have given celebrity images to the system as inputs without make up and expected from the system to find these people as output at least at top 10 ranking even if it can not find it in the first place. As it is shown in the figure 5.3-1 we have given Jennifer Lopez without make up to the system and the program found her at 1st ranking. Additionally Halle Berry’s and Courtney Cox’s pictures without make up are given to the system and the program found their pictures with makeup in the third ranking.

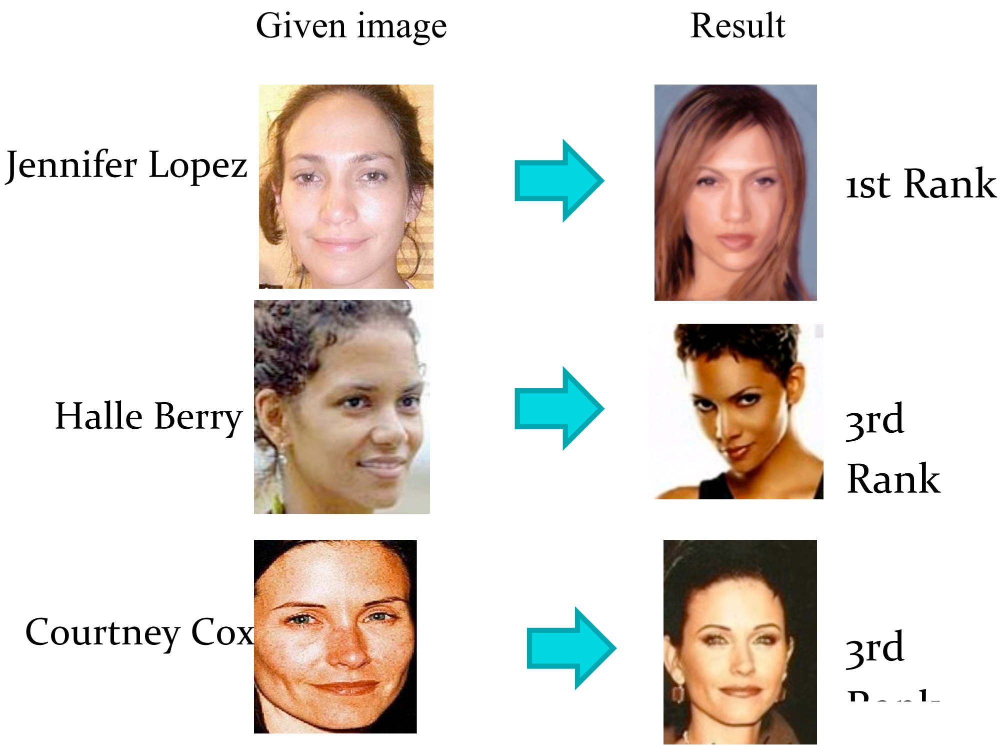


Figure 5.3-1 – Output for make-up difference test

5.4 Morph

Finally as the hardest test case of the application we used morphed images of celebrities and tried to find these two celebrities as the output of the program at the top 10 ranking. In order to achieve these morphed pictures we used a different program that takes two pictures and merges them and returns one single image as the combination of the other two. So we have given that image as input to the system. As it is shown in the figure 5.4-1 we have given the image of the morph of Jessica Alba and Angelina Jolie. As a result we received Angelina Jolie as the first rank and Jessica Alba as the seventh ranking. Moreover when we gave the morphed image of Jennifer Lopez and Jennifer Aniston we got Jennifer Lopez as the second and Jennifer Aniston as the tenth of the output list.

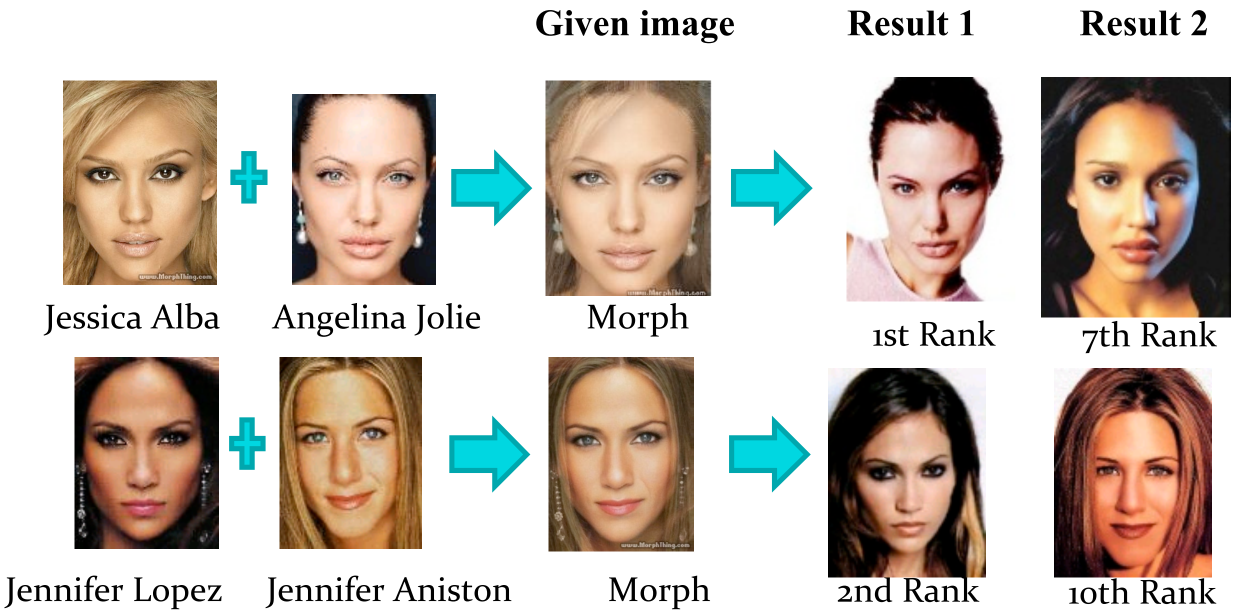


Figure 5.4-1 – Output for make-up difference test

REFERENCES

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

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

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

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

APPENDIX A – Definitions

Euclidean distance: In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler. [2] The Euclidean distance between points P = $(p_{1,}\, _{p2}, \ldots, p_n)$ and $Q = (q_1,\ q_2,\ \ldots,\ q_n)$, in Euclidean *n*-space, is defined as:

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

Interest point detection: Interest point detection is a recent terminology in computer vision that refers to the detection of interest points for subsequent processing. An interest point is a point in the image which in general can be characterized where it has a clear, preferably mathematically well-founded, definition; a well-defined *position* in image space, the local image structure around the interest point is rich in terms of local *information contents*, such that the use of interest points simplifies further processing in the vision system. It is *stable* under local and global perturbations in the image domain, including deformations as those arising from perspective transformations (sometimes reduced to affine transformations, scale changes, rotations and/or translations) as well as illumination/brightness variations, such that the interest points can be reliably computed with high degree of *reproducibility*. Optionally, the notion of interest point should include an attribute of *scale*, to make it possible to compute interest points from real-life images as well as under scale changes. [3]

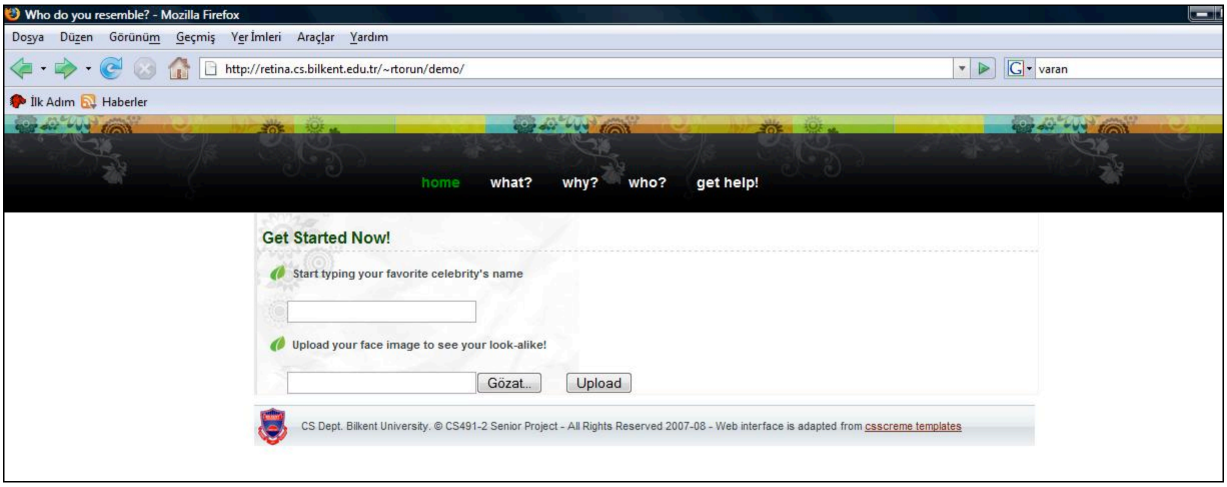
Lowe's SIFT Keypoint Detector: In order to perform feature extraction of the face images in the database, Lowe's SIFT operator is used which gives interest points of the given images. These interest points are used to perform matching between images. SIFT operator consists of 4 major stages which are Scale-space extrema detection, Keypoint localization, Orientation Assignment, and Keypoint descriptor [3].

First stage searches over all scales and image locations by using difference-of-Gaussian function. In the second stage, a detailed model is fit to determine location and scale at each candidate location. Keypoints are selected based on measures of their stability. In the third stage one or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. In the last stage of the SIFT technique the local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. [4]

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

APPENDIX B – User's Manual

The final product of our system is really easy to use. It is available on the internet from the address: retina.cs.bilkent.edu.tr/~rtorun/demo. Figure.B-1 shows the starting page:



Who do you resemble? - Mozilla Firefox

http://retina.cs.bilkent.edu.tr/~rtorun/demo/

Get Started Now!

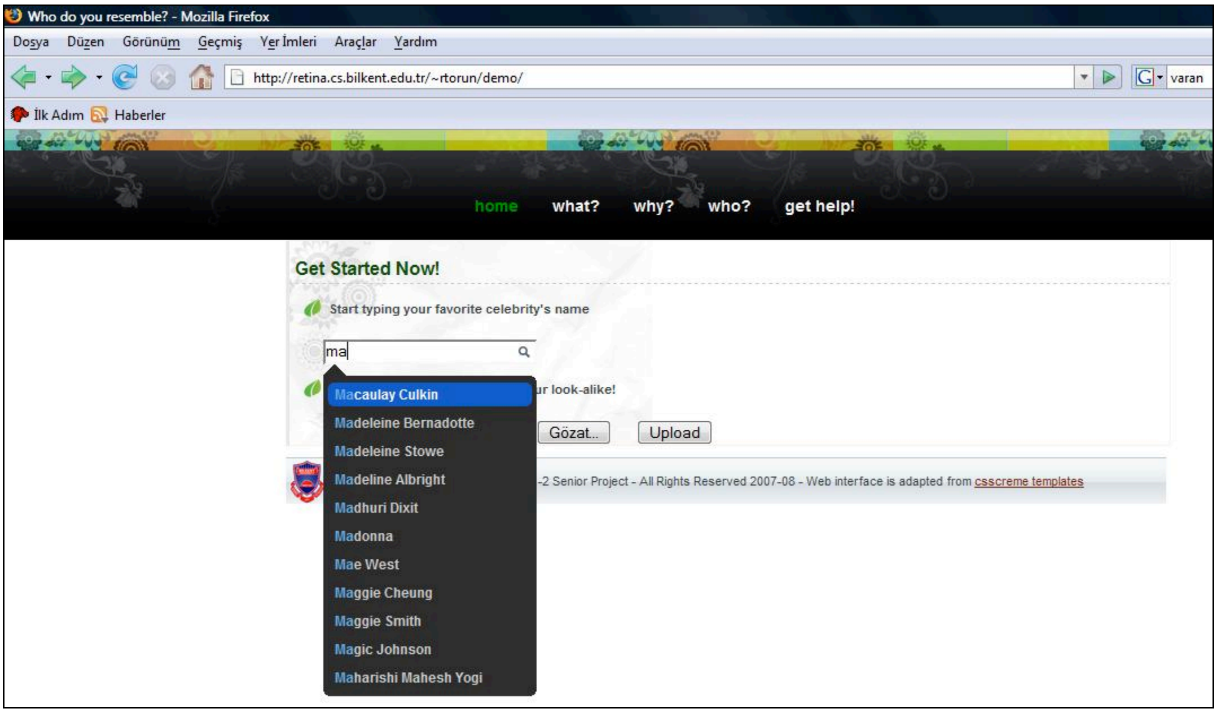
Start typing your favorite celebrity's name

Upload your face image to see your look-alike!

CS Dept. Bilkent University. © CS491-2 Senior Project - All Rights Reserved 2007-08 - Web interface is adapted from csscreme templates

Figure.B-1 - Starting Screen

The user may choose to get results for his personal photos or for celebrities. If the user wants to find to whom a specific celebrity resembles, he can either directly upload a picture of him by using the browse button, or he can choose to type the name of the celebrity in the textbox. As the user types, the textbox shows the options he has (since the programs uses the pictures in the database when "search by text" is used by the user). The user either types the whole name or chooses the name he wants from the list. This step is shown in Figure.B-2.



Who do you resemble? - Mozilla Firefox

http://retina.cs.bilkent.edu.tr/~rtorun/demo/

Get Started Now!

Start typing your favorite celebrity's name

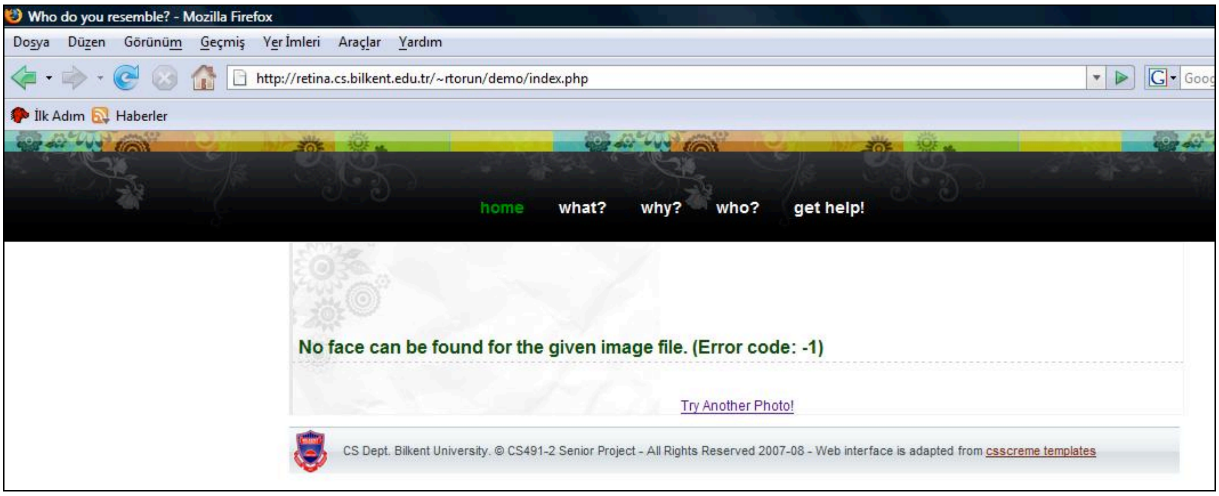
ma

ur look-alike!

-2 Senior Project - All Rights Reserved 2007-08 - Web interface is adapted from csscreme templates

Figure.B-2 – Text Based Comparison Screen

The face detection algorithm is run on the given image of the celebrity and the most similar celebrities are returned. The user has to wait for some time before he can reach the results because the given image is being compared to the entire database and in addition if the user has manually uploaded an image, then face detection algorithm is also run on the image, before comparing it to the database. If the program cannot detect the face then an error message is given as shown in Figure.B-3.



Who do you resemble? - Mozilla Firefox

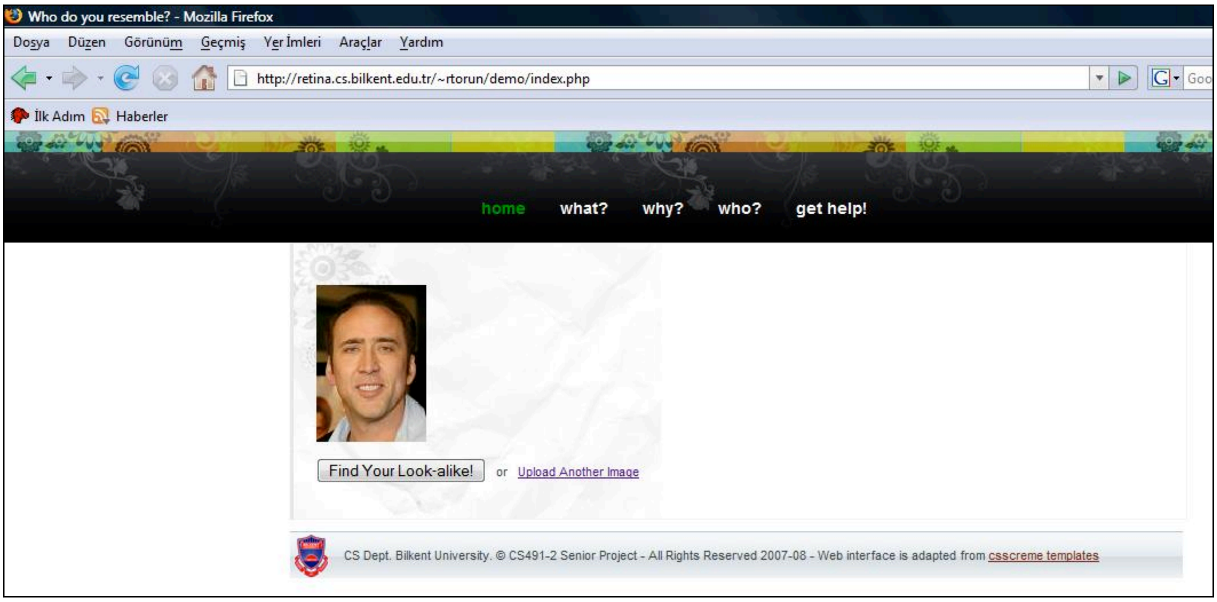
http://retina.cs.bilkent.edu.tr/~rtorun/demo/index.php

No face can be found for the given image file. (Error code: -1)

CS Dept. Bilkent University. © CS491-2 Senior Project - All Rights Reserved 2007-08 - Web interface is adapted from csscreme templates

Figure.B-3 – Error Screen

If the face is found, then one last confirmation is awaited from the user as shown in Figure.B-4.



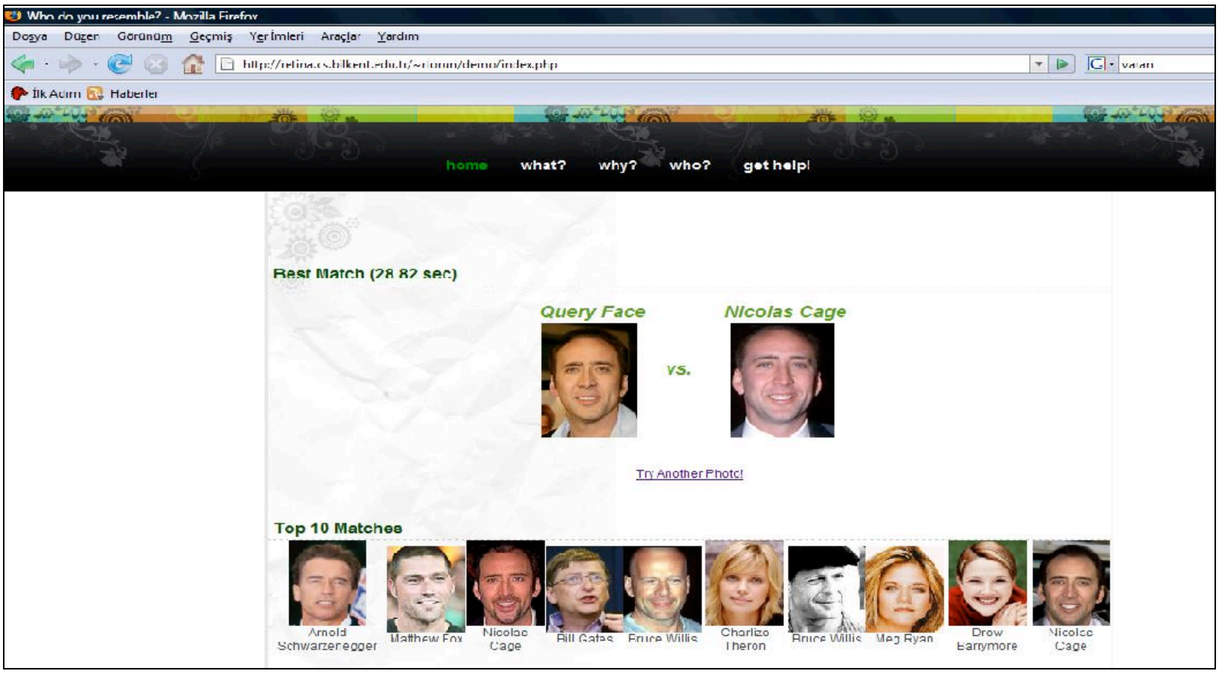
Who do you resemble? - Mozilla Firefox

http://retina.cs.bilkent.edu.tr/~rtorun/demo/index.php

CS Dept. Bilkent University. © CS491-2 Senior Project - All Rights Reserved 2007-08 - Web interface is adapted from csscreme templates

Figure.B-4 – Face Confirmation Screen

After the confirmation, the results screen shows up as in Figure.B-5. In this screen, the best match of the given image is shown near the query face and the following best 10 matches are shown below with the name of the celebrities. After seeing the results, the user can give a new query face to the system by clicking “Try Another Photo” link.


Who do you resemble? - Mozilla Firefox
Dosya Düzen Görünüm Geçmiş Yer İmleri Araçlar Yardım
http://retina.cs.bilkent.edu.tr/~roran/demo/index.php
vatan
İlk Adım Haberler
home what? why? who? get help!

Best Match (28.82 sec)

Top 10 Matches

Figure.B-5 – Results Screen

APPENDIX C – Test Charts

Person 569: With face detection

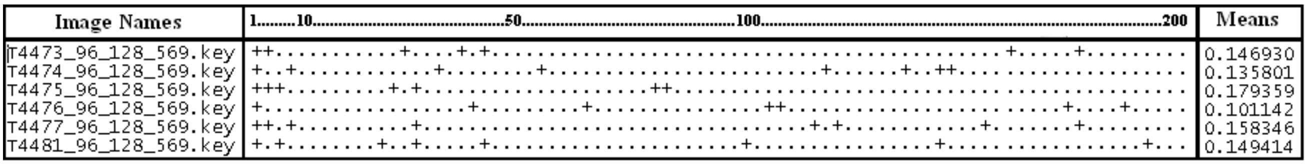


Image Names	1.........10...50...100..200	Means
T4473_96_128_569.key	++............+....+.+..+......+.........	0.146930
T4474_96_128_569.key	+..+.............+.........+.........................+.......+..++.....................	0.135801
T4475_96_128_569.key	+++..........+.+.....................++..	0.179359
T4476_96_128_569.key	+...................+..........+................++..........................+....+.....	0.101142
T4477_96_128_569.key	++.+...........+....................................+.+.............+........+.........	0.158346
T4481_96_128_569.key	+.+.........+..+.....+........................+.................+..................+...	0.149414

Person 569: Without face detection

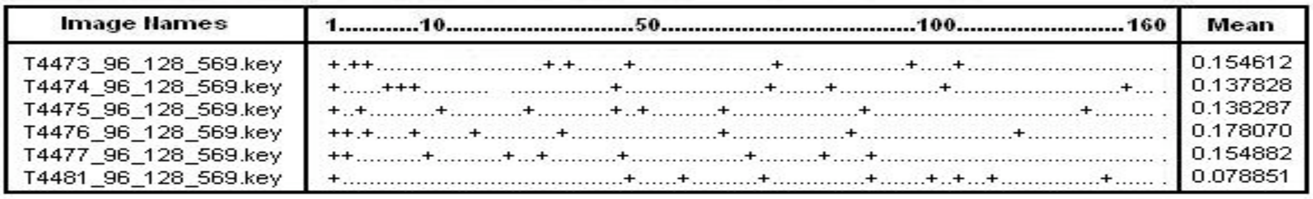


Image Names	1...........10...........................50....................................100.........................160	Mean
T4473_96_128_569.key	+.++........................+.+.........+......................+....................+.......+............................. .	0.154612
T4474_96_128_569.key	+.....+++.......... ....+......................+.........+.................+...........................+... .	0.137828
T4475_96_128_569.key	+..+...........+.............+.............+..+...........+.....................+.................................+.......... .	0.138287
T4476_96_128_569.key	++.+.....+.......+.............+........................+...................+.........................+.................... .	0.178070
T4477_96_128_569.key	++............+............+....+............+...................+..........+.......+.. .	0.154882
T4481_96_128_569.key	+..+........+............+................+.........+.+..+.................+....... .	0.078851

Person 681: With face detection

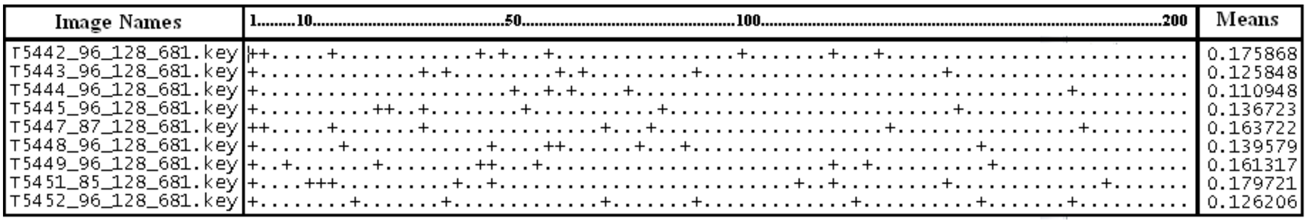


Image Names	1.........10...50...100..200	Means
T5442_96_128_681.key	++.....+.............+.+...+.................+........+...+............................	0.175868
T5443_96_128_681.key	+..............+.+..........+.+..........+......................+......................	0.125848
T5444_96_128_681.key	+.......................+..+.+.....+..+..........	0.110948
T5445_96_128_681.key	+..........++..+.........+............+..........................+.....................	0.136723
T5447_87_128_681.key	++.....+.......+................+....+.....................+.................+.........	0.163722
T5448_96_128_681.key	+.......+.............+....++.......+..+...........................+...................	0.139579
T5449_96_128_681.key	+..+.......+.........++...+...........................+..+..........+..................	0.161317
T5451_85_128_681.key	+...+++............+..+............................+..+.........+..............+.......	0.179721
T5452_96_128_681.key	+........+........+.............+........+..............+..........+........+..........	0.126206

Person 681: Without face detection

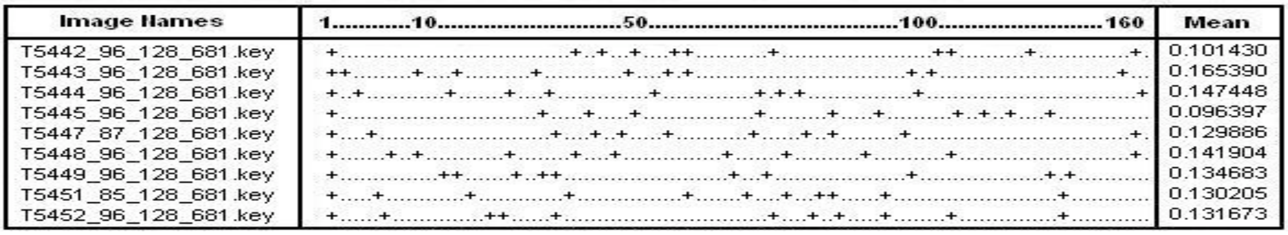


Image Names	1...........10...........................50....................................100.........................160	Mean
T5442_96_128_681.key	+.....................................+.+..+...++............+.........................++............+...............+.	0.101430
T5443_96_128_681.key	++...........+.....+............+..............+.....+..+..................................+.+..............................+..	0.165390
T5444_96_128_681.key	+..+..............+.........+.....+................+................+.+.+..................+..................................+	0.147448
T5445_96_128_681.key	+................................+.......+.......+...................+...........+.......+............+..+..+.....+..............	0.096397
T5447_87_128_681.key	+.....+............................+......+...+......+.............+.......+...+...........+...+.	0.129886
T5448_96_128_681.key	+........+...+..............+..........+.....+.................+.........+............+.............+............................+.	0.141904
T5449_96_128_681.key	+................++.......+..++..........................+...+......................+....................+.+............	0.134683
T5451_85_128_681.key	+......+.............+...............+..................+.........+....+...++........+...........................+.............	0.130205
T5452_96_128_681.key	+.......+..............++.......+..................................+.....+...+.......+.........+.................+............	0.131673

Person 770: With face detection

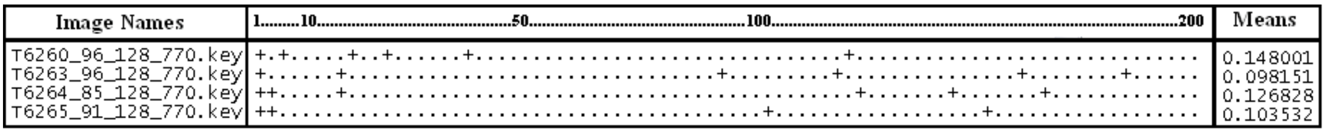


Image Names	1.........10...50...100..200	Means
T6260_96_128_770.key	+.+.....+..+.......+..................................+................................	0.148001
T6263_96_128_770.key	+......+..................................+..........+................+........+.......	0.098151
T6264_85_128_770.key	++.....+...+........+.......+..............	0.126828
T6265_91_128_770.key	++...+...................+...................	0.103532

Person 770: Without face detection

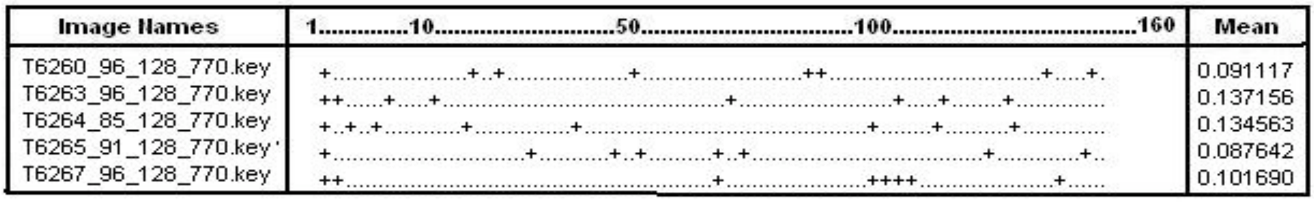


Image Names	1.............10...........................50................................100.......................................160	Mean
T6260_96_128_770.key	+......................+..+....................+.........................++.................................+.....+.	0.091117
T6263_96_128_770.key	++.....+.....+..+.........................+......+........+..............	0.137156
T6264_85_128_770.key	+..+..+.............+.................+..+.........+...........+...............	0.134563
T6265_91_128_770.key'	+...............................+............+..+...........+..+......................................+..............+..	0.087642
T6267_96_128_770.key	++...+......................++++.....................+......	0.101690