Technique Report

Automatic Face Recognition and Annotation for Mobile Devices

Student: Chung-Lun Lu (逯仲倫)

Advisor: Dr. Been-Chian Chien (錢炳全　博士)

Database and Information Knowledge System Laboratory

Department of Computer Science and Information Engineering

National University of Tainan

October 2009

Tainan, Taiwan, R.O.C.

Automatic Face Recognition and Annotation for Mobile Devices

Student: Chung-Lun Lu (逯仲倫)

Advisor: Dr. Been-Chian Chien (錢炳全　博士)

Database and Information Knowledge System Laboratory

Department of Computer Science and Information Engineering

National University of Tainan

A Technique Report

October 2009

Tainan, Taiwan, R.O.C.

Contents

Automatic Face Recognition and Annotation for Mobile Devices

Abstract

In this report, a new feature-based face recognition approach is proposed. We first extract 11 feature points from the detected area of face. Then a human face is characterized by 24 normalized distances and 6 angles. After the characterization of faces, different classifiers, such as C4.5, Naïve Bayes, SVM, etc, are used to learn the classification model for recognizing the sets of faces. The experiments referred to the Georgia Tech face database; in particular, 5 subjects without glasses, 10 images per each subject, are selected.

Keywords: face recognition, machine learning, facial features extraction

1. Introduction

With the rapid growth of digital technology, digital cameras have replaced traditional ones using films; even mobile phones have equipped built-in cameras. The popularization of digital cameras and mobile devices bring us to the digital life. The low-priced memory promotes the digital photo's storage and its application. However, it results in problems of generating, storing and managing on a great deal of images. Hence, if photos can be annotated automatically while they were being captured, it will be helpful to manage and store for a large amount of images.

From semantic aspect, a photo can be described via four factors: people, time, place and event. Time is labeled via the camera's build-in clock; event consists of serial photos; and a 3G camera phone can determine the place by its GPS information. In this research, we focus on how to annotate the person whom is in photos.

For annotating photos automatically, one of the most important things is to recognize human in photos. A generic face recognition system proposed by Zhao et al. [12] consists of three parts: face detection, feature extraction and face recognition, see Figure 1. Face detection localizes faces from photos, feature extraction extracts some features which can represent a face and face recognition identifies the face. For part 1, we use a face recognition method which works on color images of people without wearing glasses and are scale independency. Even the images of people with moustache don't affect the extraction of facial feature points. But there are some problems need to be solved, such as pose dependency. So far, it only works well on frontal faces. Besides, images of people wearing glasses are another problem; it results in the wrong extraction of eyebrow points. For part 2, a machine-learning based approach is used. An initial model is created by few training photos; once new photos are obtained, the model grows. With the increase of input photos, the model grows and we expect that the recognition rate would increase.

General face recognition is still a difficult problem. Besides, we observed that camera users always take photos on a particular group, such as family or friends. Hence we reduce the problem on a particular group. We selected 5 people which are without glasses from the Georgia Tech face database [4] to test our system. "The Georgia Tech face database contains 50 people and all people are represented by 15 color JPEG images with cluttered background and different poses, facial expressions, lighting conditions and scale."

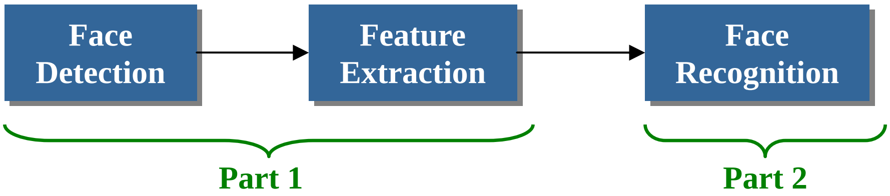


Figure 1. A generic face recognition system.

2. Review of Related Researches

Approaches of face recognition for still images can be categorized into three categories: holistic matching methods, feature-based matching methods and hybrid methods.

For holistic matching methods, whole face is input to a face recognition system. Example approaches are eigenfaces, fisherfaces, feature lines, support vector machines, etc. Liao and Li [8] used a Gabor-based complex vector to present facial features and proposed two approaches, two-layer nearest neighbor approach and modular nearest feature line approach for face recognition and they obtained 95.5 % and 96.0 % recognition rates respectively. An integration of moment invariant and PCA for face recognition is proposed by Phiasai et al. [11].

For feature-based matching methods, local facial features are extracted and then input to a classifier. Example approaches are hidden Markov model and convolution neural network. Nefian et al. [2] used a HMM-based method, instead of extracting the exact locations of features, the eye, nose, mouth, and chin are covered by strips of pixels, for face detection and recognition and obtained 86.0% recognition rate. Lawrence et al. [10] presented a system which combined local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network and resulted in 3.8% error rate. Wiskott et al. [6] developed a graph matching system which extracted image graph based on the bunch graph for face recognition, and on different training and testing sets, the recognition rates range from 57.0% to 98.0%.

For hybrid methods, both local facial features and the whole face are used to recognize a face. Example approach is modular eigenfaces. Pentland et al. [1] proposed a view-based multiple-observer eigenspace technique for face recognition under variable pose and 95.0% recognition rate is obtained. Some recognition techniques and results are shown in Table 1.

Table 1. Some face recognition techniques.

<table>
<tr><th>Approach</th><th>Training Set</th><th>Testing Set</th><th>Image Database</th><th>Image Type</th><th>Recognition Rate</th></tr>
<tr><td>PCA + Moment Invariant [11]</td><td>- 10 persons
- 1 image per person</td><td>- 10 persons
- 9 image per person</td><td>ORL database</td><td>- Different scales, lights, poses, and expressions
- Resolution: 112*92 pixels</td><td>94.0% ~ 96.0%</td></tr>
<tr><td>Two-Layer Nearest Neighbor (TLNN) [8]</td><td rowspan="2">- 40 persons
- 5 images per person</td><td rowspan="2">- 40 persons
- 5 images per person</td><td rowspan="2">Cambridge database</td><td rowspan="2">- Different viewpoints within a person class
- Different races, ages and, genders between person classes
- All frontal faces
- Resolution: 128*128 pixels</td><td>95.5%</td></tr>
<tr><td>Modular Nearest Feature Line (MNFL) [8]</td><td>96.0%</td></tr>
<tr><td>Hidden Markov Model (HMM) [2]</td><td>- 40 persons
- 5 images per person</td><td>- 40 persons
- 5 images per person</td><td>ORL database</td><td>- Different facial expressions, hair styles, and eye wear
- Resolution: 112*92 pixels</td><td>86.0%</td></tr>
<tr><td>Convolutional Neural Network [10]</td><td>- 40 persons
- 5 images per person</td><td>- 40 persons
- 5 images per person</td><td>-</td><td>- Different expressions, poses, and facial details</td><td>96.2%</td></tr>
<tr><td rowspan="6">Elastic Bunch Graph Matching [6]</td><td>- 250 persons
- 1 neutral frontal image per person</td><td>- 250 persons
- 1 different facial expression image per person</td><td rowspan="3">FERET database</td><td rowspan="3">- Different poses: frontal, quarter view, half-profile right or left (rotated 40-70^{o}), and profile right or left
- Images are labeled with pose
- Resolution: 256*284 pixels</td><td>98.0%</td></tr>
<tr><td>- 250 persons
- 1half-profiles and right rotated image per person</td><td>- 180 persons
- 1half-profiles and left rotated image per person</td><td>57.0%</td></tr>
<tr><td>- 250 persons
- 1 profiles and right rotated image per person</td><td>- 250 persons
- 1 profiles and left rotated image per person</td><td>84.0%</td></tr>
<tr><td rowspan="3">- 108 persons
- 1 neutral frontal image per person</td><td>- 108 persons
- 1 different facial expression image per person</td><td rowspan="3">Bochum database</td><td rowspan="3">- Different poses: frontal, rotated 11^{o} and 22^{o}
- Images are labeled with pose</td><td>91.0%</td></tr>
<tr><td>- 108 persons
- 1 face rotated 11^{o} image per person</td><td>94.0%</td></tr>
<tr><td>- 108 persons
- 1 face rotated 22^{o} image per person</td><td>88.0%</td></tr>
<tr><td>Modular Eigenfaces [1]</td><td>- 45 persons
- 1 image with neutral facial expression per person</td><td>- 45 persons
- 1 image with smiling facial expression per person</td><td>-</td><td>-</td><td>95.0%</td></tr>
</table>

3. The Architecture of Face Recognition

The architecture of face recognition is illustrated in Figure 2 and outlined as follows.

- *Face Detector* detects and localizes human faces in the input photos.
- *Facial Features Extractor* extracts some facial feature points from the detected faces and characterizes a face via these extracted points.
- *Machine Learning* and *Classifier* are in charge of face recognition.

Faces are detected and localized while photos input. Once faces are localized, facial features extraction is applied and 11 feature points are extracted. After that, a face is characterized by 6 angles and 24 normalized distances, and the training data are generated. Then classifier and machine learning provide many different classification algorithms on the training data to recognize faces.

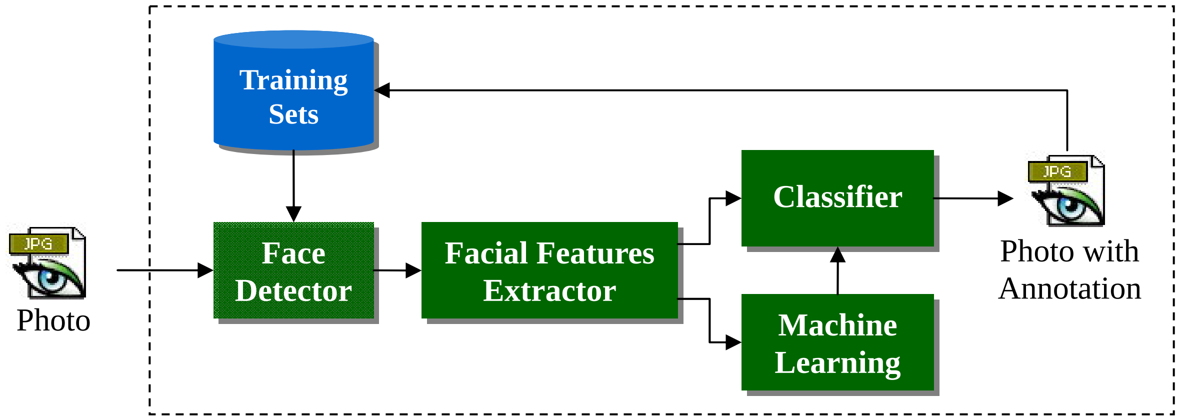


Figure 2. Architecture of Face recognition.

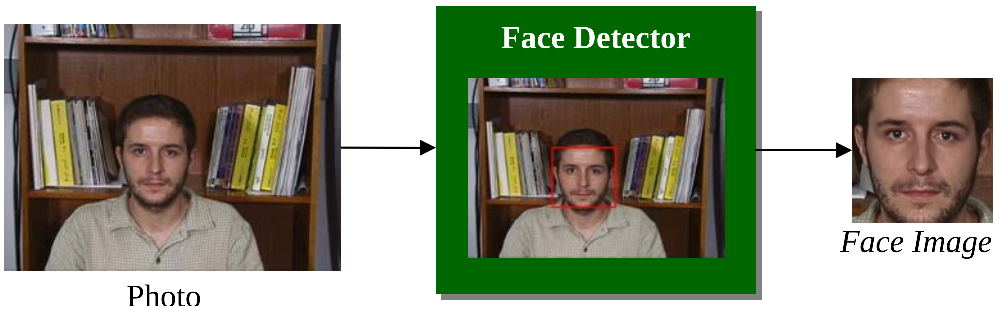


Figure 3. The result of the face detector.

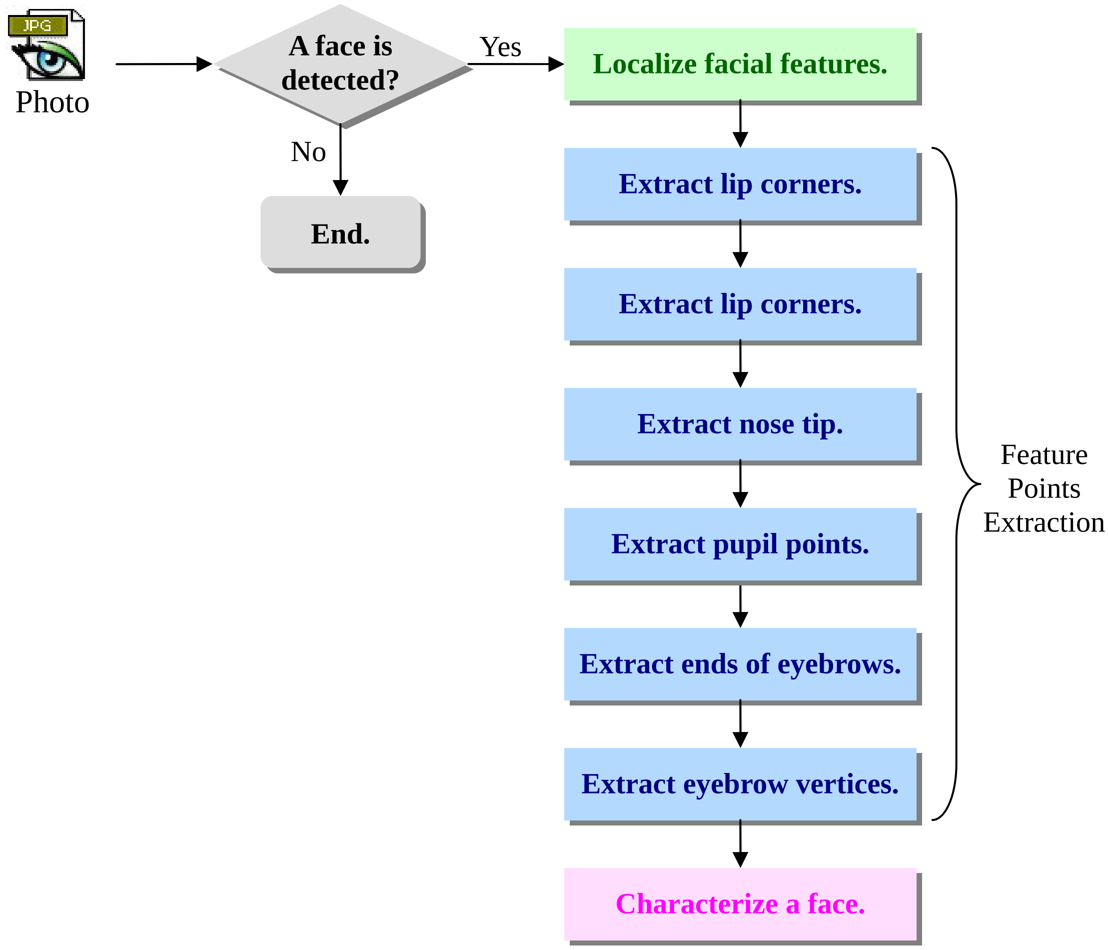


Figure 4. The algorithm of facial features extraction.

4. The Features Extraction and Face Recognition Methods

The face detector in the architecture of face recognition uses Intel's OpenCV [7]. Figure 3 shows the detected result.

The algorithm of facial features extraction is illustrated in Figure 4. After the face image is obtained from face detector, input it to facial features extractor. This facial features extractor consists of three subsequent modules. First, it localizes mouth, nose, eyes and eyebrows which are known as facial features. Second, it extracts 11 feature points and eyebrow shapes. Third, it characterizes a face by 6 angles and 24 normalized distances.

4.1 Facial Features Localization

Figure 5 shows that how a face image is segmented to localize facial features. Assume that the width and the height of a face image are w and h respectively. The face image is segmented into three areas: the mouth area, the nose area and the eye and eyebrow area. The width of the mouth area (green rectangle) is $0.5 \times w$ and the height is $0.2 \times h$, and its start point is (0.25×*w*, 0.7×*h*). The width of the nose area (blue rectangle) is 0.2×*w* and the height is 0.25×*h*, and its start point is (0.4×*w*, 0.47×*h*). The width of the eye and eyebrow area (red rectangle) is 0.7×*w* and the height is 0.3×*h*, and its start point is (0.15×*w*, 0.2×*h*). Beside these three areas, an eyebrow area (top half of red rectangle) is also segmented.

Figure 6 shows the segmentation results. For our subsequent modules, when extracting the feature points, the search area is restricted to the corresponding area instead of segmenting a face image into four parts. For instance, if we want to find the point of nose tip, search the blue area described in Figure 5 instead of the whole face image.

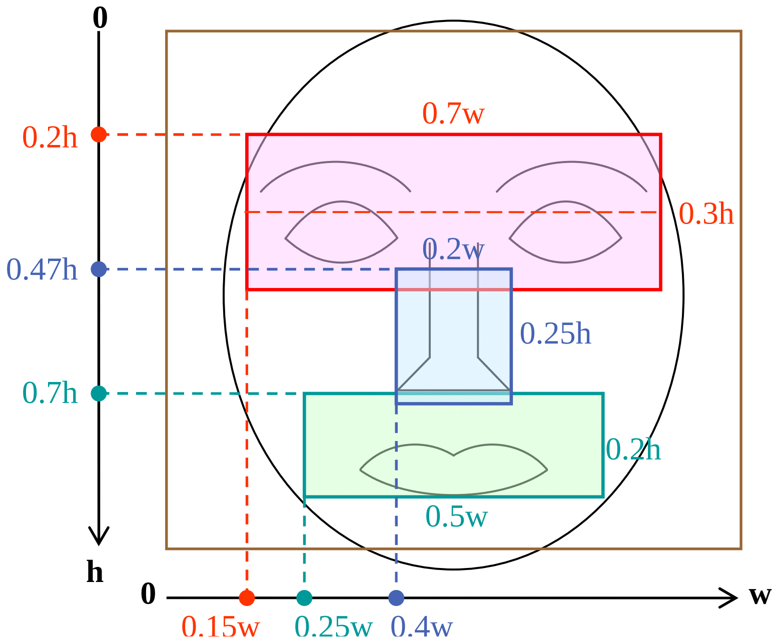
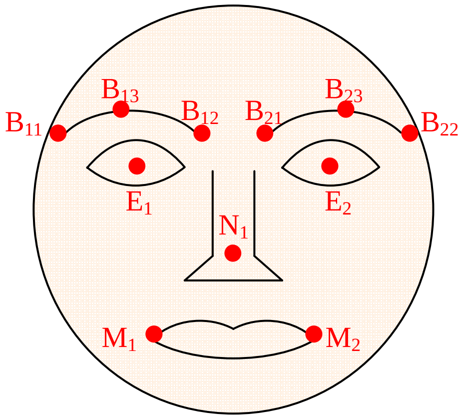


Figure 5. Facial features localization.

Figure 6. The results of the Facial features localization module.

4.2 Extraction of Feature Points

The extraction module determines 11 feature points per face image: the lip corners, the nose tip, the pupils, ends of eyebrows and the eyebrow vertices. Figure 7 shows the 11 feature points.



- M_1 and M_2: lip corners
- N_1: nose tip
- E_1 and E_2: pupils
- B_{11}, B_{12}, B_{21} and B_{22}: ends of eyebrows
- B_{13} and B_{23}: eyebrow vertices

Figure 7. The 11 feature points.

4.2.1 Mouth

To find the lip corners, we build a mouth map, *MouthMap*, which is proposed in Hsu et al. [9] "The color of mouth region contains stronger red component and weaker blue component that other facial regions. Hence, the chrominance component C_r is greater than C_b in the mouth region. And the mouth has a relatively low response in the C_r/C_b feature, but it has a high response in $(C_r)^2$." The *MouthMap* is constructed as follows:

$$MouthMap = (C_r)^2 \cdot ((C_r)^2 - \eta \cdot C_r / C_b)^2,$$

$$\eta = 0.95 \cdot \frac{\sum_{(x,y)\in FI} C_r(x,y)^2}{\sum_{(x,y)\in FI} (C_r(x,y)/C_b(x,y))},$$

where $(C_r)^2$ and C_r / C_b are normalized to [0,255] and *FI* is the face image obtained from face detector. The *MouthMap* is then binarized: pixels outside the mouth area are all set to 0 and 10% of the highest values in the mouth area are set to 1. Before extracting the lip points, we cluster the binarized *MouthMap* to reduce the noise which may be generated because of the relative high threshold when binarizing. Finally the points of lip corners are determined by taking the most lateral extremes. Figure 8 illustrates the lip corners extraction procedure. Figure 9 shows some results of lip points extraction.

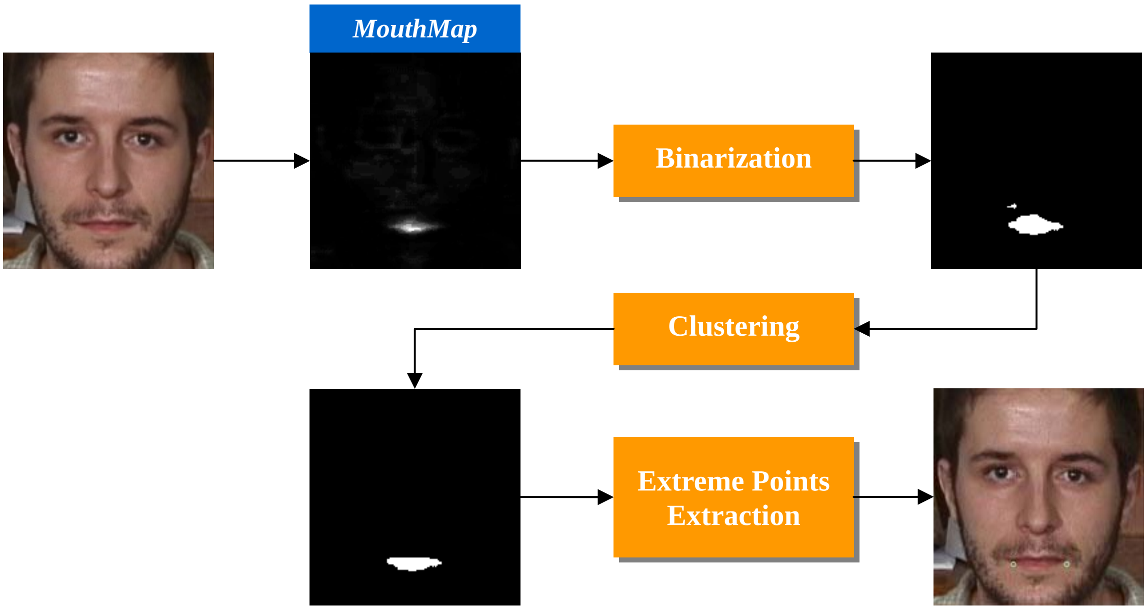


Figure 8. The procedure of lip points extraction.

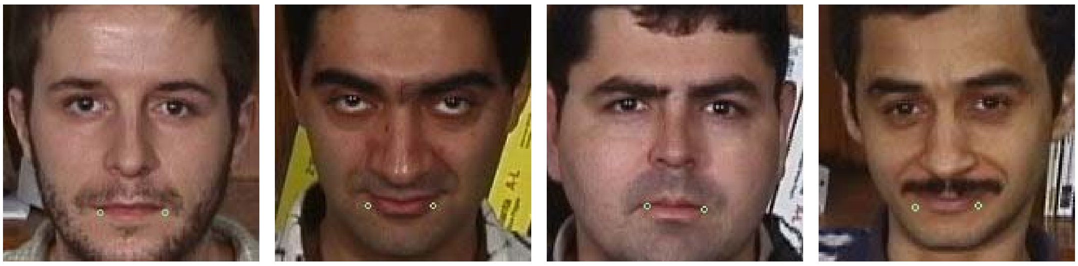


Figure 9. The results of lip corners extraction.

4.2.2 Nose

The point of nose tip is found by a very simple strategy. We consider the nose tip has the brightest gray level. So just search the nose area and find the point which has the brightest gray level. Figure 10 shows some results of nose point extraction.


Figure 10. The results of nose point extraction.

4.2.3 Pupils

In order to find the pupil points, we first build an eye map, *EyeMap,* which is proposed in Hsu et al. [9]. The *EyeMap* consists of two eye maps: *EyeMapC* and *EyeMapL*. *EyeMapC* is from the chrominance components and *EyeMapL* is from the luminance component. The *EyeMapC* is constructed by

$$EyeMapC = \{(C_b(x,y))^2 + (\tilde{C}_r(x,y))^2 + (C_b(x,y)/C_r(x,y))\}/3,$$

where $(C_b)^2$, $(\tilde{C}_r)^2$ and C_b/C_r are normalized to [0,255] and $\tilde{C}_r$ is the negative of C_r. The *EyeMapL* is constructed by

$$EyeMapL = \frac{Y(x,y) \oplus s(x,y)}{Y(x,y)\Theta s(x,y) + 1},$$

where $\oplus$ is dilation operation, Θ is erosion operation and *s* is a structuring element shown in Figure 12. *EyeMapC* and *EyeMapL* are combined into *EyeMap* by an AND operation,

$$EyeMap = EyeMapC(x,y)\ \mathbf{AND}\ EyeMapL(x,y).$$

After getting the *EyeMap*, we operate dilation on it to get *EyeMapD*. To extract boundary, we first erode *EyeMapD* by the structuring element *s* and then perform the difference between *EyeMapD* and its erosion. The result is named *EyeMapB*. That is,

$$EyeMapB = EyeMapD(x,y) - (EyeMapD(x,y)\Theta s(x,y)).$$

At final, we separate the *EyeMapB* into two equal parts: left and right parts. For each part we find its central point by averaging the coordinate of all white pixels. The central point in the left part is the pupil of left eye; the central point in the right part is the pupil of right eye. Figure 11 illustrates the pupil extraction procedure. Figure 13 shows some results of pupil points extraction.

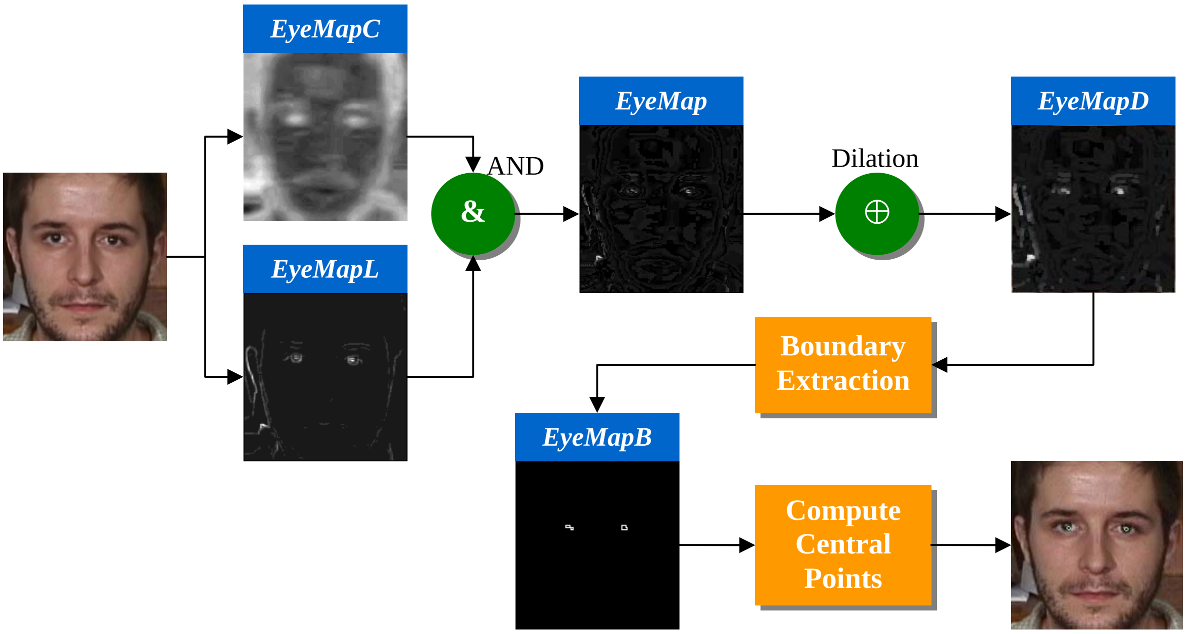


Figure 11. The procedure of pupil extraction.

1	1	1
1	1	1
1	1	1

Figure 12. Structuring element *s*.

Figure 13. The results of pupil extraction.

4.2.4 Eyebrows

To extract the eyebrow vertices and its ends, an image, denoted as *FeatureImage*, is built. The *FeatureImage* is obtained from the following subsequent steps. First, we apply the Prewitt filter to the input photo to extract edge. Second, the edge image is contrasted to highlight its edge. Then we binarize the contrasted image and apply a 5×5 median filter to it to reduce noise. Figure 14 illustrates the procedure to generate *FeatureImage*.

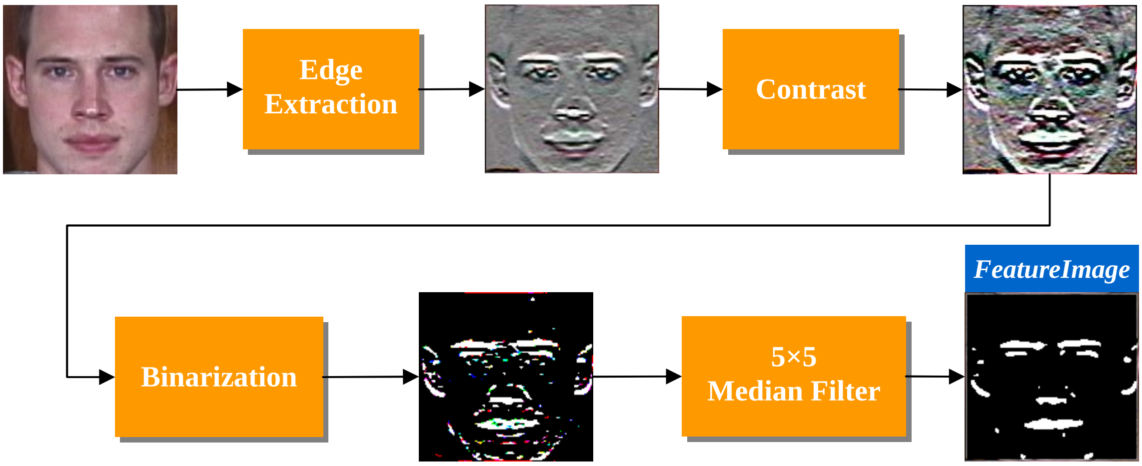


Figure 14. The procedure to generate *Feature Image*.

After the *FeatureImage* is obtained, extract the eyebrow points. At first, we build a binarized image which will be used to disable pixels outside the eye and eyebrow area and also disable some noise inside the eye and eyebrow area. For every pixels p in the face image, p is set to 1 if its value is large than the average value; otherwise, it set to 0. Then combine the binarized image (BI) with *FeatureImage* (FI) using the following algorithm:

Input: FI(resolution: $m \times n$ pixels), BI(resolution: $m \times n$ pixels)

Output: A combined image, CI, with resolution $m \times n$ pixels

Step 1. For every pixels in FI, Test to see if FI(x, y) is outside the eye and eyebrow area.
If it is, assign 0 to CI(x, y) and go to step 4; otherwise, go to step 2.

Step 2. Test to see if FI(x, y) equals to 0 and BI(x, y) equals to 1.
If it is, assign 1 to CI(x, y) and go to step 4; otherwise, go to step 3.

Step 3. Assign 0 to CI(x, y) and go to step 4.

Step 4. Move (x, y) to the next pixel in FI and go to step 1.

Algorithm 1. The algorithm to generate the combined image.

The combined image, *CImage*, is used to determine the eyebrow ends by taking the most lateral extremes and determine the eyebrow vertices by taking the top extremes. Figure 15 illustrates the procedure of eyebrow points extraction. Figure 16 shows the results of eyebrow points extraction (eyebrow ends: green points, eyebrow vertices: yellow points).

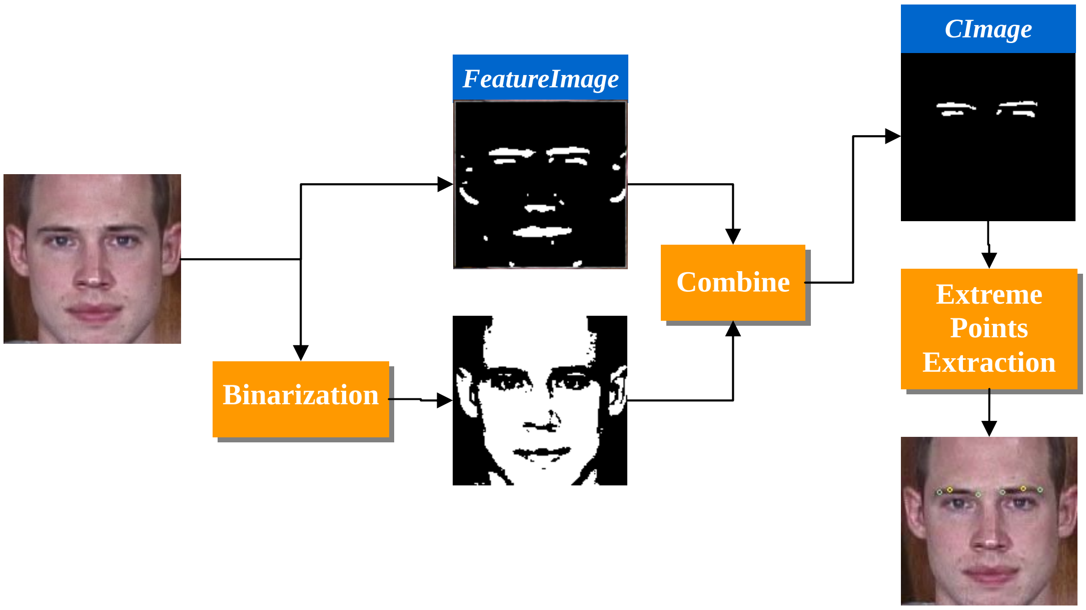


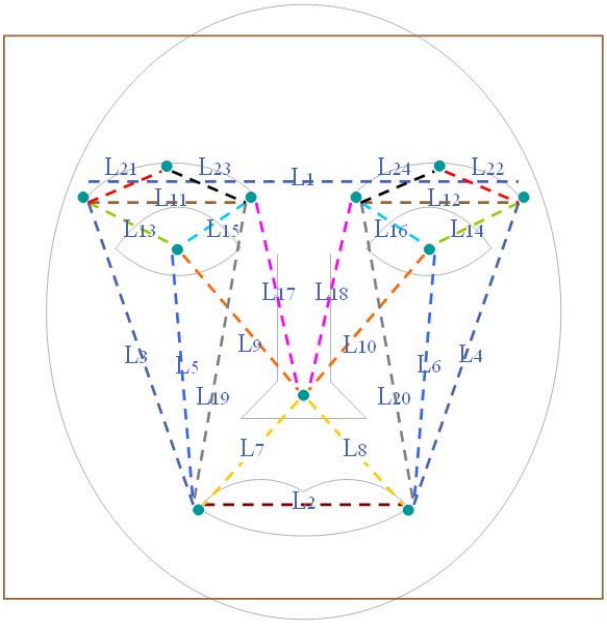
Figure 15. The procedure of eyebrow points extraction.

Figure 16. The results of eyebrow points extraction.

4.3 Face Characterization

A face is characterized by 6 angles and 24 distances. The 24 distances are all normalized, the unit for normalization is $\overline{E_1E_2}$. Figure 17 shows the 24 distances.

For each eyebrow, we extract 3 points, two for the eyebrow ends and one for its vertex. These three points construct a triangle. The triangle's interior angles are added to characterize a face.



L1: $\overline{B_{11}B_{22}}$
L2: $\overline{M_1M_2}$
L3: $\overline{B_{11}M_1}$
L4: $\overline{B_{22}M_2}$
L5: $\overline{E_1M_1}$
L6: $\overline{E_2M_2}$
L7: $\overline{N_1M_1}$
L8: $\overline{N_1M_2}$
L9: $\overline{E_1N_1}$
L10: $\overline{E_2N_1}$
L11: $\overline{B_{11}B_{12}}$
L12: $\overline{B_{21}B_{22}}$
L13: $\overline{B_{11}E_1}$
L14: $\overline{B_{22}E_2}$
L15: $\overline{B_{12}E_1}$
L16: $\overline{B_{21}E_2}$
L17: $\overline{B_{12}N_1}$
L18: $\overline{B_{21}N_1}$
L19: $\overline{B_{12}M_1}$
L20: $\overline{B_{21}M_2}$
L21: $\overline{B_{13}B_{11}}$
L22: $\overline{B_{22}B_{23}}$
L23: $\overline{B_{12}B_{13}}$
L24: $\overline{B_{23}B_{21}}$

Figure 17. The 24 distances for face characterization.

5. System Implementation and Experimental Results

5.1 System Implementation

Face recognition system (FRS) is a software for facial feature points extraction. Main features of FRS include GUI Interface, C# source, and extraction of Mouth, nose, eye and eyebrow points.

Some snapshots of FRS are shown from Figure 18 to Figure 25.

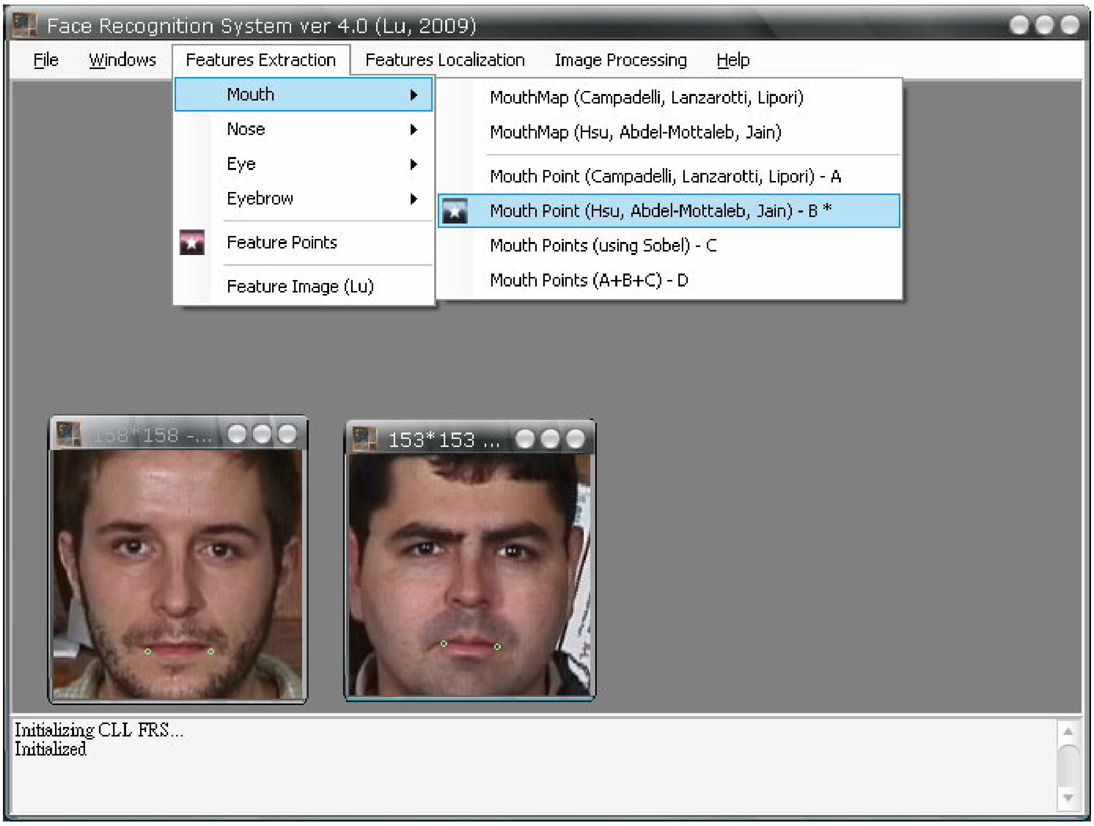


Figure 18. Mouth points extraction of FRS.

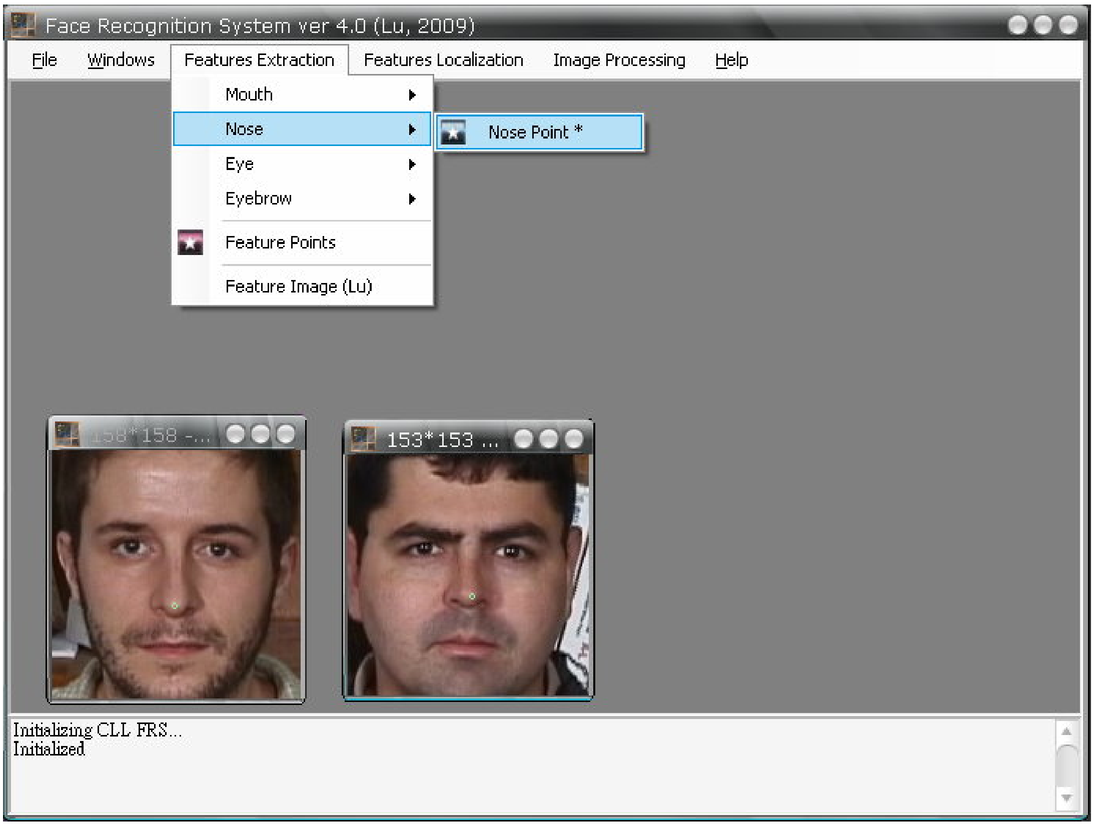


Figure 19. Nose point extraction of FRS.

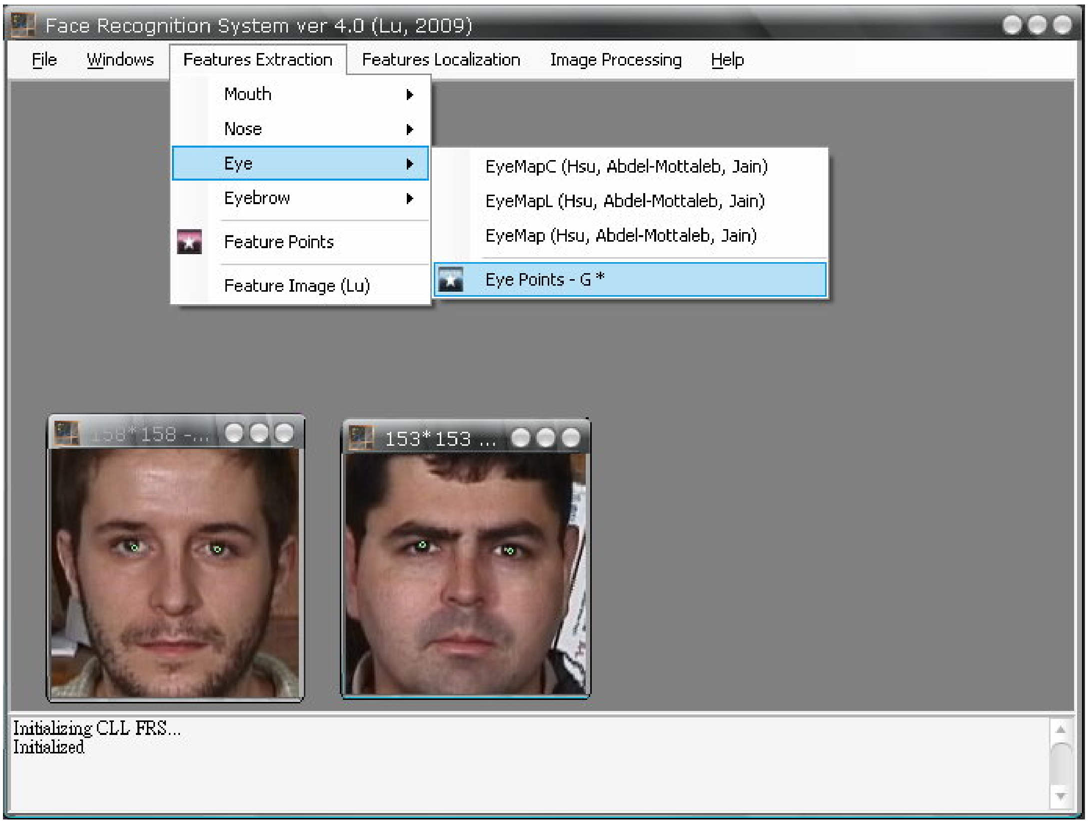


Figure 20. Eye points extraction of FRS.

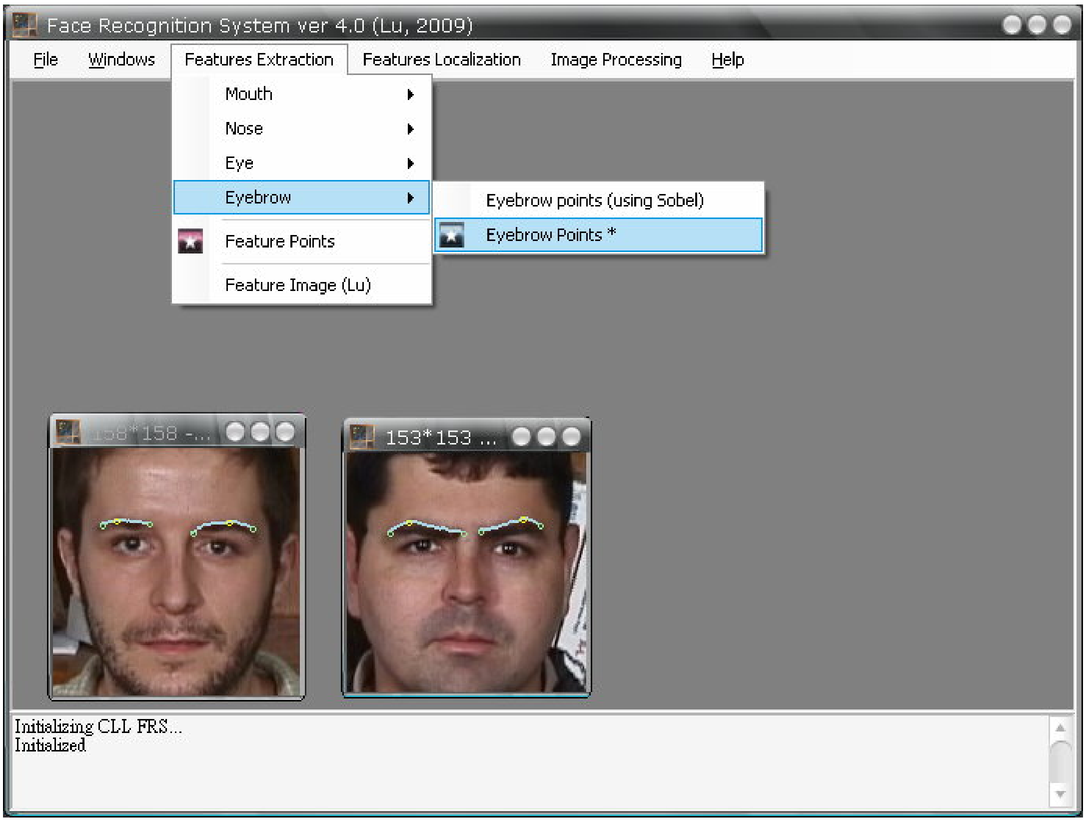


Figure 21. Eyebrow points extraction of FRS.

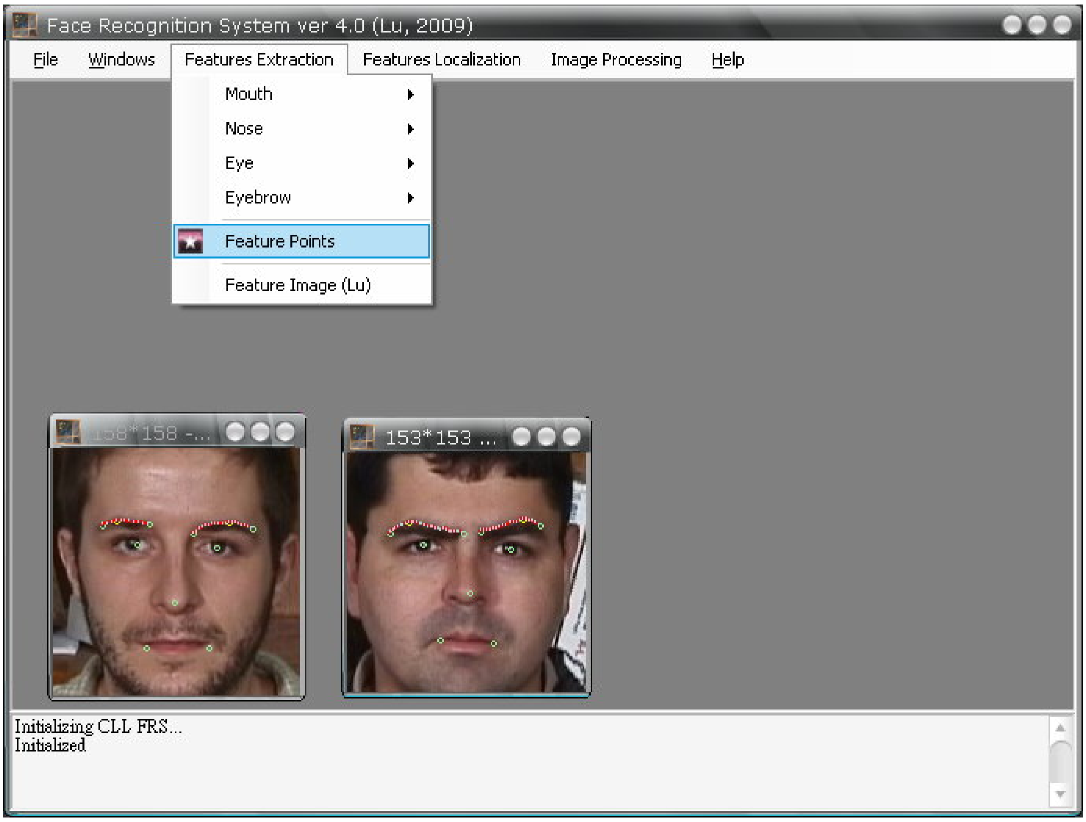


Figure 22. Feature points extraction of FRS.

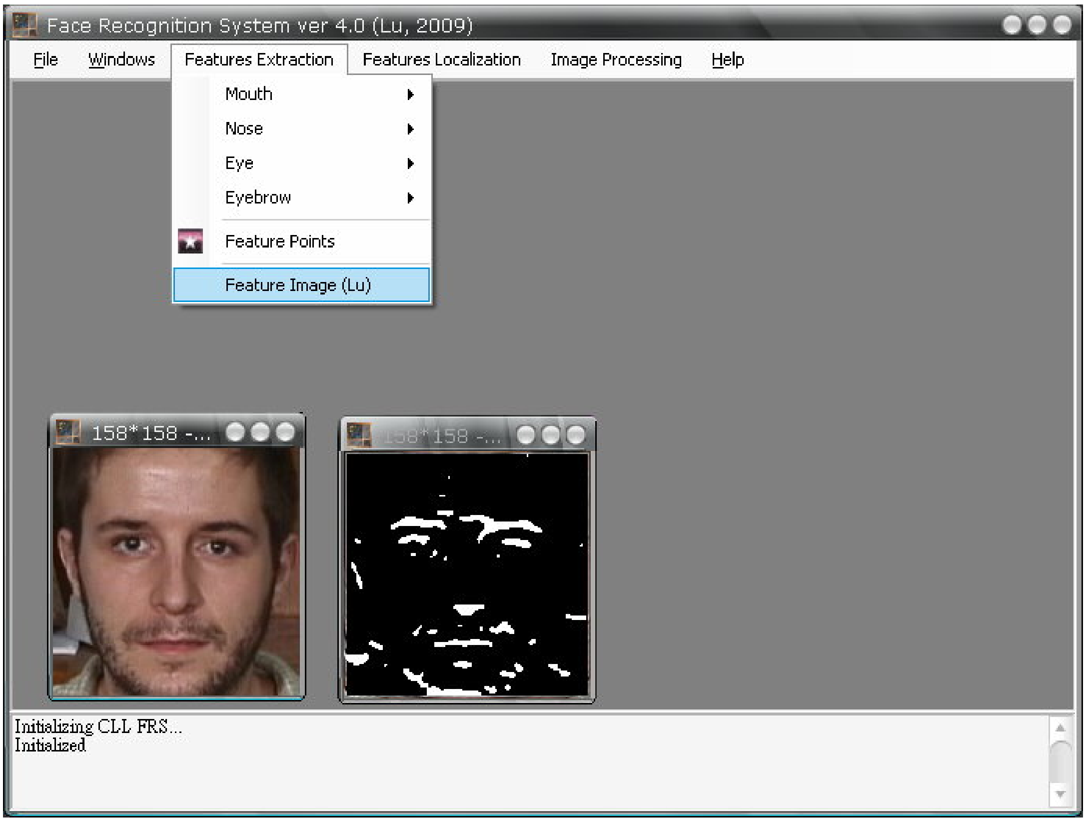


Figure 23. Feature image generation of FRS.

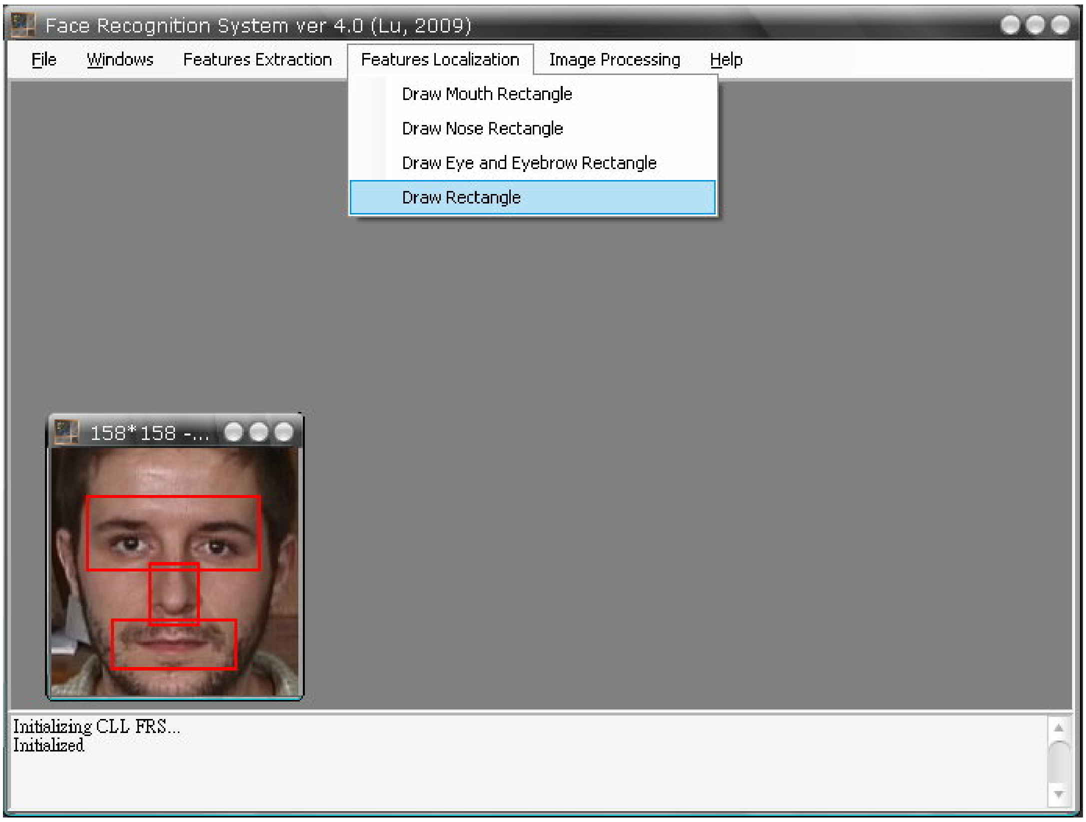


Figure 24. Facial features segmentation of FRS.

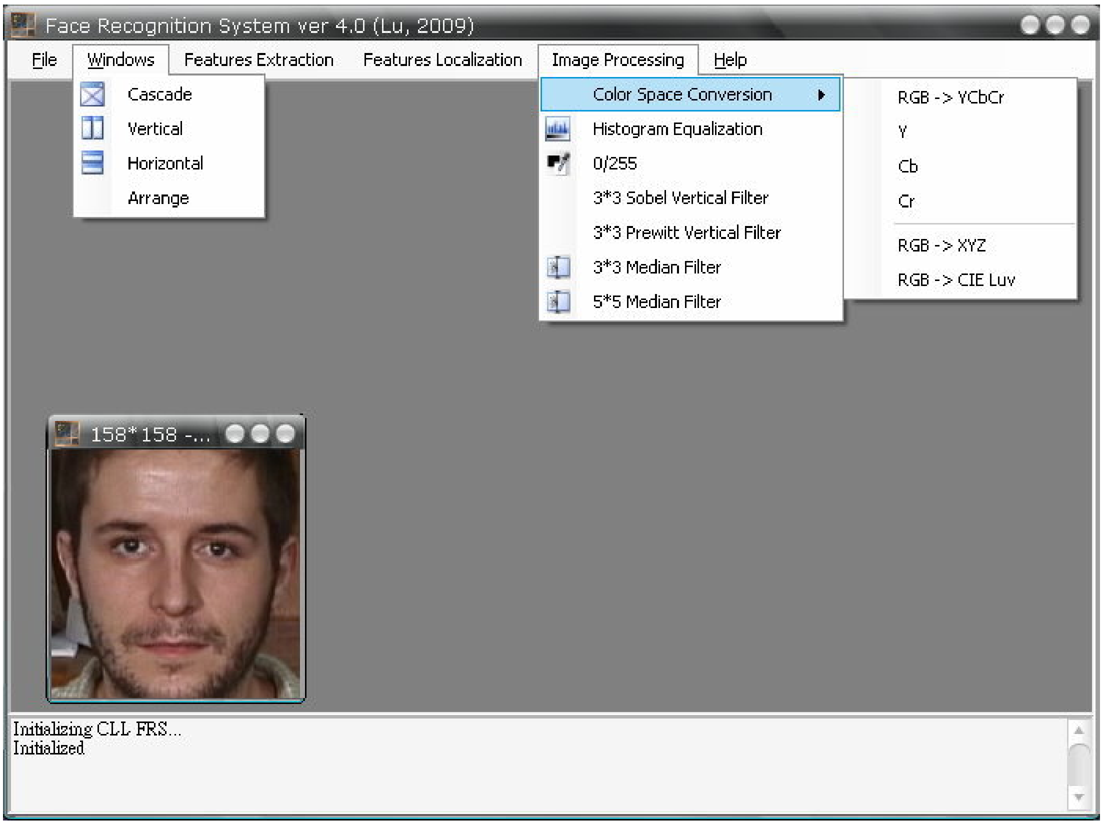


Figure 25. Other functions.

5.2 Experimental Results

To evaluate the performance, different algorithms (SVM [3], C4.5, Naïve Bayes, RBF Network, etc [5] classifiers) are used. The experiment referred to the Georgia Tech face database; in particular we use the images of 5 subjects without glasses, 10 images per each subject. Figure 26 shows these 5 subjects and the obtained results are reported in Table 2.

Figure 26. The 5 subjects in our training data.

Table 2. The experimental results.

	Test Mode			
	Use training set	Cross validation (10 folds)	Cross validation (5 folds)	Cross validation (2 folds)
C4.5	96%	62%	56%	70%
NB Tree	100%	64%	70%	62%
LAD Tree	100%	78%	72%	48%
DB Tree	78%	60%	56%	50%
FT	100%	90%	82%	20%
LMT	100%	86%	86%	72%
FLR	100%	68%	72%	64%
SMO	94%	86%	80%	76%
Random Forest	100%	66%	70%	72%
Simple Logistic	100%	86%	86%	72%
Logistic	100%	88%	88%	76%
Multilayer Perceptron	100%	86%	88%	80%
Naïve Bayes	96%	74%	74%	70%
RBF Network	100%	68%	60%	58%
LibSVM	76%	70%	70%	62%

6. Discussion and Future Work

According to the results, the recognition rate can be affected by the used classifier, so determining a suitable classifier is important. Besides, due to our application, we expect that the recognition rate would increase while the input photos increase; hence, some classifiers, such as functional tree (FT), logistic model tree (LMT), sequential minimal optimization (SMO) , logistic, and Naïve bayes, may be our choices.

So far, we characterize a face by 24 distances and 6 angles and the recognition is based on the off-line learning. In our future research, we will incorporate 4 angles and the eyebrow shape into the system to improve the recognition rate since from observation, the eyebrow shape and its rotated angles are significant features to distinguish persons. The 4 angles are used to describe the relative positions between pupils and eyebrow ends, and the eyebrow shape will be obtained by clustering. Besides, on-line learning will be used on recognition.

References

[1] A. Pentland, B. Moghaddam, and T. Starner, “View-Based and Modular Eigenspaces for Face Recognition”, in Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 84-91, 1994.

[2] A. V. Nefian and M. H. Hayes III, “Face Detection and Recognition Using Hidden Markov Models”, in Proc. of IEEE International Conference on image Processing, vol. 1, pp. 141-145, 1998.

[3] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] Georgia Tech Face Database, http://www.anefian.com/research/face_reco.htm

[5] I. H. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques”, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[6] L. Wiskott, J. –M. Fellous, N. Kuiger, C. von der Malsburg, “Face Recognition By Elastic Bunch Graph Matching”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, pp. 775-779, 1997.

[7] Open Computer Vision Library (OpenCV), http://sourceforge.net/projects/opencvlibrary/

[8] R. Liao and S. Z. Li, “Face Recognition Based on Multiple facial Features”, in Proc. of 4th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 239-244, 2000.

[9] R. L. Hsu, M. Adbel-Mottaleb, and A, K, Jain, “Face Detection in Color Image”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, 2002.

[10] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face Recognition: A Convolutional Neural-Network Approach”, IEEE Trans. On Neural Networks, vol.8, pp. 98-113, 1997.

[11] T. Phiasai, S. Arunrungrusmi, and K. Chamnongthai, “Face Recognition System With PCA and Moment Invariant Method”, IEEE International Symposium on Circuits and Systems, vol. 2, pp. 165-168, 2001.

[12] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face Recognition: A Literature Survey”. ACM, Computing Surveys, vol. 35, no. 4, pp. 399-458, 2003.

Appendix

Following is the image gallery we used for experiment (taken from Georgia Tech face database).