City University of Hong Kong

Department of Computer Science

06CS046

Project Title

Face Detection and Face Recognition of Human-like Characters in Comics

(Volume 1 of 1)

Student Name : Savina Cheung

Student No. :

Programme Code : BSCCS

Supervisor : Dr. LEUNG, Wing Ho Howard

1st Reader : Dr. NGO, Chong Wah

2nd Reader : Prof. IP, Ho Shing Horace

For Official Use Only

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Face Detection and Face Recognition of Human-like Characters in Comics

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: Savina Cheung Chui Shan   Signature: ______________

Student ID: ______________   Date: 2007/4/14

Abstract

In a nutshell, it is inconvenient for comic readers to perform a scene search on large volumes of comic pages, as a conventional way to achieve the task is to perform brute force searching based on the vague impression of searchers. With the emergence of e-comics, computers could be designed to achieve the search task by comic characters indexing. The search of characters under different occasions will be helpful in identifying which scenes are the craved ones by narrowing down the scope from the large amount of digital comic pages in the database. To be able to differentiate between various cartoon characters for indexing, a content based image retrieval (CBIR) system is developed for the sake of comic readers. Under this project several detection and recognition strategies would be investigated to determine which algorithms, when being applied on e-comic data set, are more workable. After the comparison on the workable face detection and recognition algorithms were done from the literature, some of them have been culled to experiment on the comic data set. Overall 7 algorithms (3 for detection and 4 for recognition) are selected to work on the experiments, and the most workable methodologies are found to be Adaboost (detection) and Elastic Bunch Graph Matching [EBGM](recognition), yielding a rate of 45.50% and 54.44% respectively. To compensate for the imperfectness of the detection rate, the CBIR system developed are embedded with a modification function for users to add in undetected faces as for input in recognition; where to improve the recognition result, some knowledge from the comic nature are utilized as to boost the performance of EBGM, resulting an increase of 38.79% from the original recognition rate, the overall recognition first-rank rate is finalized as 75.50%. Although the performance is still not 100% accurate, the CBIR system might be able to search the specific scene if users provide more information to it. The CBIR system deployed is

also designed in such a way that, if being used continuously, the performance of recognition will be enhanced.

Acknowledgement

First and foremost, my deepest gratitude goes to my supervisor, Dr LEUNG, Wing Ho Howard, for his guidance throughout the entire year. Being amazed by his values towards this project, I believe that this kind of attitude should be complimented. Without this sort of directions, definitely I would not have possessed adequate vigor to complete the task but would have relinquished for long. Apart from the technological potion of the project, he also put much emphasis on the presentation performance, of which I am weak at. He also exhibited great extend of patience in supervising this project when I encountered obstacles. Furthermore, his suggestions on the design and improvement to my application not only enhance the system itself, but also open my eyes to be aware of how current applications are designed. All of these come to my personal and the project's benefit. Sincerely, I am grateful to have his yearlong supervision and advice.

By this opportunity I would also like to extend my appreciation to the Department of Computer Science of City University of Hong Kong, especially for those professors and tutors whom I have acquired knowledge from for all these years. Without such fundamental cognition it would be far more difficult to comprehend and progress on the development of the project.

Last but not least, thank is to be expressed to kingcomics.com, for providing the data set in digital form that is a critical factor to the completion of this project.

Content

Chapter 1-- Introduction

1.1 The Problem

The most important content in comics is the plot of the story, which is the premier intention for comic readers to purchase and enjoy them. Comic characters are always an essential element in the creation of a narrative. So story plots and characters are always adhered to each other.

With the aid of technology, comic readers now can obtain their favorite comics in electronic form. And as the accessibility of electronic comics is becoming increasingly handy, there is a trend of reading comics on PC rather than on the traditional printed paper volumes.

However, it is a common nature of comics to be distributed in hundreds of volumes, resulting in thousands of comic pages. In addition, sometimes it takes quite a while for the publishers to distribute the next volume. Along with another property of comics, being that the plot can always be related to a scene that happens in a far earlier volume; the three factors are quite troublesome for comic lovers, especially those who are following an active comic rather than a retired one. It is indeed a rather tricky task to find out what had happened in previous chapters if the comic readers forget some details or want to find the correlation between chapters.

1.2 The Solution

As images reside in readers' mind more than text and they tend to find a particular scene with certain characters by exhaustive search, a superior comic indexing approach is to search comic pages in terms of different comic image characters rather than simply by text. Identifying which character to which is vital in searching a particular scene from the comics because the characters are always the main theme of the scene. Having the information of where the characters are located, the search of finding a particular scene can then be tapered down, and hopefully it is more efficient for comic readers to perform a scene search. This project explores the possibility of applying existing face detection and recognition technology based on content based image retrieval (CBIR) to build a system for identifying individual comic characters among a set of digital comic images.

1.3 Project Scope

1.3.1 Content-based image retrieval (CBIR)

Content-based image retrieval (CBIR) is currently an active research area in the computer vision community. Unfortunately, there are only few CBIR systems that can handle e-comics. All of the data of e-comics are available as multimedia documents, i.e. documents consisting of different types of data such as text and images. However, little work has been done on content-based image retrieval to specifically handle digital comics.

In this project a CBIR system which demonstrates face detection and recognition techniques to allow the retrieval of comic images from queries of comic characters will be presented. As the CBIR system is mainly built on comic characters detection and recognition, the detection and recognition of comic characters will be the main scope.

1.3.2 Face Detection and Recognition

As there are numerous face detection and recognition methods could be used for detecting comic characters from comic images, the project will focus on investigating which of them is more promising in bringing a better performance in such comic searches. After the algorithms which would work on comic image sets have been identified, further modifications on the algorithm may be proposed in order to improve the result of the identification process of comic characters. These methods will be discussed in a later section.

1.3.3 Data Set

In a research point of view, numerous of face detection and recognition have been done on registered images, for example, a very common dataset used by researchers to perform their experiments is the FERET Dataset. And many algorithms are able to perform a high accuracy on this kind of dataset.

Figure 1.3.3 Some examples from the FERET Dataset obtained from [35]

However, if the dataset are not that registered, the performances of the algorithms might be not such powerful. It is never possible for comic images to be perfectly registered to suit in the algorithms. Thus, along with developing the CBIR system to cater for comic users' crave, in this paper we would also like to investigate which existing techniques are more invariant from the pose, expression and of a given face image.

Below lists the different types of faces that we would like the face detector to be able to detect from the given set of comic images:

- Frontal and Rotated Frontal

- Profile

- Non-skin color

- Rotated by 90 degrees

- Occluded

1.4 Organization of Following Sections

First a brief review of some existing face detection and recognition algorithms will be provided in Chapter 2. In the consequent chapter, the more feasible algorithms for the project will be identified and be described in detail. The algorithms will then be applied to build a comic character search application and it will be introduced in Chapter 4. Subsequently, the experiments had been done to investigate the performance of different methodologies implemented will be revealed in Chapter 5. The final part of the paper will present the conclusions.

Chapter 2-- Literature Review

This section discloses some of the popular face detection and recognition algorithms which has been proposed by other researchers.

2.1 Overview of Face Detection and Recognition Stage

Face detection and recognition has been an active research topic in computer vision for more than two decades. Here are the key tasks to be performed:

- Face detection (localization). It detects where the faces are located.
- Facial feature extraction. Key face features from the faces such as eyes, mouth, chin, are extracted to undergo recognition or tracking.
- Face recognition. A stage of matching a facial image to a reference image existed in the training data.
- Face authentication or verification. A positive or negative reply will be given to determine whether a new facial image matches with the reference ones.

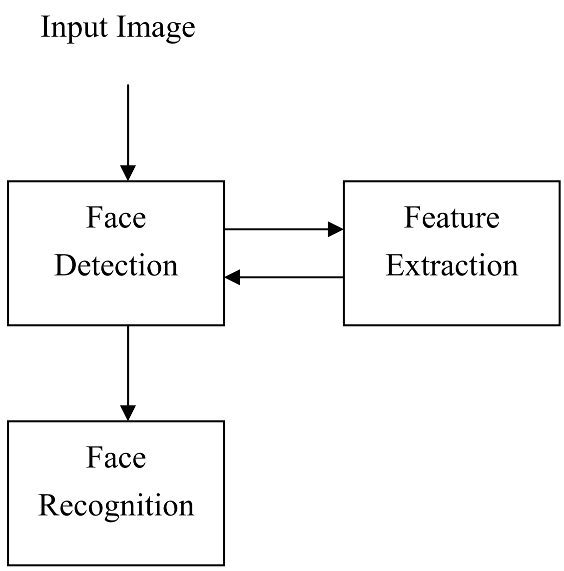


Figure 2.1. *Configuration of Face Recognition System*

A detail outline of different algorithms of both detection and recognition will be presented in the Chapter 3 so as to compare our scenario with the nature of the algorithms.

2.2 Face Detection

Face detection methods are often classified into 2 main categories in *Figure 2.2*: Feature Based Approaches and Image Based Approaches [1].

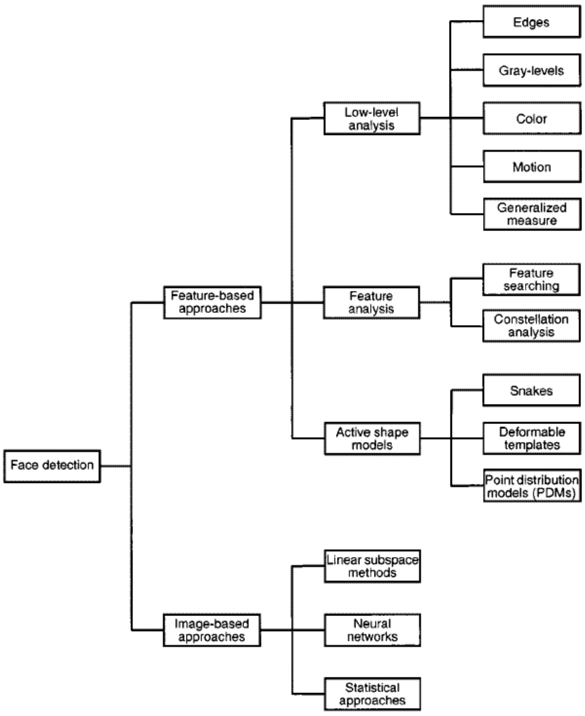


Figure 2.2 *Classification of Face Detection Methodologies [1]*

2.2.1 Feature-Based Approach

Feature based approaches include methods based on edges, lines, and curves. Basically depend on structural matching with textural and geometrical constraints.

For instance, in edge representation, which was applied by Sakai et al. [2], works by drawing face lining from images to locate facial features.

Using a slightly different feature from curves and lines, De Silva et al. [3] carried out their detection study which started by scanning the image from top to bottom, and at the same time searched for the top of a head and then a sudden increase in edge densities, which indicates the location of a pair of eyes to detect whether there is a face in the given image.

2.2.1.1 Low-level Analysis

Low-level analysis deals with the segmentation of visual features using pixel properties such as gray-scale and color.

Because of the low-level nature, features generated from this analysis are ambiguous, as we make our goal at higher accuracy, we may consider some other approaches that can generate more explicit features.

2.2.1.2 Feature Analysis

In feature analysis, visual features are organized into a more global concept of face and facial features using information of face geometry. Through feature analysis, feature ambiguities are reduced and locations of the face and facial features are determined. Features are invariant to pose and orientation change.

Facial features are difficult to locate because of corruption such as illumination, noise, and occlusion. Also it is difficult to detect features in complex background.

2.2.1.3 Active shape models

Models have been developed for the purpose of complex and non-rigid feature extraction such as eye pupil and lip tracking. Active shape models depict the actual physical and hence higher-level appearance of features. Once released within a close proximity to a feature, an active shape model will interact with local image features (edges, brightness) and gradually deform to take the shape of the feature.

This method is simple to apply; however, templates need to be initialized near the face images or it won't work, and as the main idea is template matching, it is impossible to enumerate templates for different poses.

2.2.2. Image-Based Approach

Face detection by explicit modeling of facial features has been troubled by the unpredictability of face appearance and environmental conditions. Although some of the recent feature-based attempts have improved the ability to cope with the unpredictability, most are still limited to head, shoulder and part of frontal faces. There is still a need for techniques that can perform in more hostile scenarios such as detecting multiple faces with clutter-intensive backgrounds.

Image-based approaches ignoring the basic knowledge of the face generally work by recognizing face patterns from a set of given images, mostly known as the training stage in the detection method. After this initial stage of training, the programs may be able to detect faces which are similar to the face pattern from an input image. Comparison of distance between these classes and a 2D intensity array extracted from an input image allows the decision of face existence to be made.

Most of the image-based approaches apply a window scanning technique for detecting faces. The window-scanning algorithm is merely an exhaustive search of the input image for possible face locations at all scales.

An example of these approaches involves linear subspace method such as principal component analysis (PCA) and linear discriminant analysis (LDA). It functions by expressing the principal component of face distribution by eigenvectors. When this analysis is done, each training face can be represented as a linear component of largest eigenvectors, forming eigenfaces [4].

Applying a different technique in image-based approaches, Rowley et al. [5] adopt a Neural network approach which trained by using multiple multilayer perceptrons with different receptive fields. Then merging is done on the overlapping detections within one network. An arbitration network has been trained to combine the results from different networks. This neural network approach is also classified as image-based approach because it works by identifying face patterns.

2.3 Face Recognition

There has been numerous face recognition methods developed over the past years.

Some proposed face recognition methods recognize faces by extracting features. One of them completes the task by a template-based approach [6]. Templates are introduced to detect eyes and mouth in images. An energy function is defined that links edges in the image intensity to corresponding with the properties in the template.

The Active Shape Model proposed by Cootes et al.[7] is more flexible than the template-based approach because " the advantages using the so-called analysis through synthesis approach come from the fact that the solution is constrained by a flexible statistical model"[8].

According to Lu [9], face recognition algorithms can be classified into appearance-based and model-based approach.

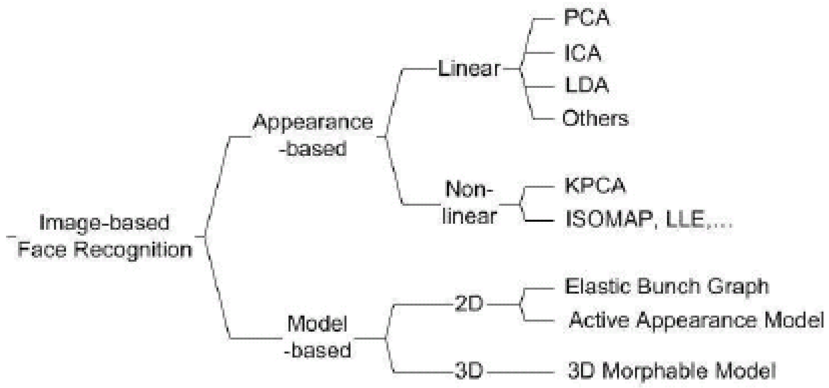


Figure 2.3 *Classification of Face Recognition Methodologies [9]*

2.3.1 Appearance-Based Approach

It is based on object views. It applies statistical techniques to analyze distribution of object image vectors and derive a feature space accordingly.

2.3.2 Model-Based Approach

Elastic Bunch Graph Matching

Wiskott et al. [10], making use of geometry of local features, proposed a structural matching category named as Elastic Bunch Graph Matching (EBGM). They used Gabor wavelets and a graph consisting of nodes and edges to represent a face. With the face graph, the model is invariant to distortion, scaling, rotation and pose.

3D morphable model

Blanz et al.[11] proposed that face recognition can be achieved by encoding shape and texture in terms of model parameters in order to build a 3D morphable model which can handle different face expressions and poses. And recognition is done by finding similarity between the query image and the prototype of this architecture.

Chapter 3 -- Methodologies for Experiments

In this section 3 detection (Skin Color Segmentation, Adaboost and Neural Network) and 4 face recognition methodologies (PCA, LDA, Bayesian Classifier and EBGM) which are to be experimented on the comic data set will be dwelled on.

3.1 Face Detection

3.1.1 Specialty to be considered for Comic Set

The main purpose for the face detection stage in our application is for preparing the face recognition stage. Provided with the ground truth tool, faces can be located manually by users; but this is often time consuming. With the help of face detection, faces can be located automatically and hopefully it can decrease the time locating all the faces by hand. Thus the following criteria are being considered for the choice of face detection methodologies:

- Accuracy

  For accuracy, it is likely for the results to have both false detected and miss faces. Since false detect and miss are dependent on each other (if the false detection rate is high then the miss rate will be lower; and vice versa), high false detection rate over high miss rate is preferred as it is more efficient for users to delete a false face rather than re-locating a missing face.

- Localization

  Locating the exact region of the faces (but not quasi ones) is crucial such that the key features of the face should be included but not any other unnecessary features.

3.1.2 Skin Color Segmentation

Since the bulk of the face images are of skin color, a direct method to determine where faces are located at could be as simple as looking for the pixel value of the comic page to see which of them lies under the skin color threshold [12][13].

To get the best result for skin color detection, firstly the color space which could provide the best representation of skin color has to be chosen (*Figure 3.1.2b*). Then the threshold is obtained by sampling under a lot of face images which appear as skin color.

Afterwards, segmentation is done and the "to-be" faces of which the pixels value lies under the determined threshold will be extracted out (*Figure 3.1.2c*). As some of the pixels, even lies within the threshold, will not be a face; to remove the scatters which will not possibly be a face, erosion is perform (*Figure 3.1.2d*); and after erosion some of the "to-be" faces will be shrunken, in turn affecting the localization of the result, thus after erosion is done dilation will be carried out (*Figure 3.1.2e*).

Finally the blobs can be identified by opting out the inside blobs of a larger blob (*Figure 3.1.2f*).

Figure 3.1.2a

Figure 3.1.2b

Figure 3.1.2c

Figure 3.1.2d

Figure 3.1.2e

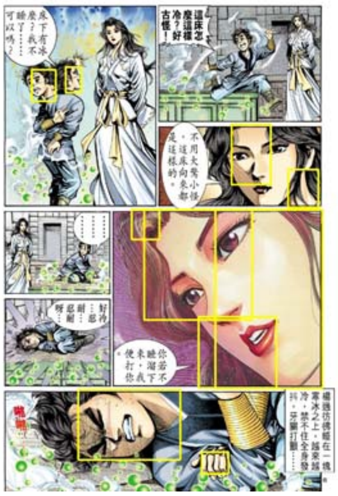


Figure 3.1.2f

Figure 3.1.2 *The procedures of skin color segmentation and blob finding*

Apparently, by just using the skin color segmentation there will be a lot of false positives. To improve the result the detected blobs which are too narrow (absolutely will not have a face contained) would be filtered away. The blobs which do not have more than 2 dark regions on the top half of the blob and without any dark regions on the bottom half of the blob, which assumes to corresponds to the 2 eyes and the mouth, are thrown away and not counted to be a detected face (the eyes and mouths are marked on the faces on *Table 5.1.2.2*).

3.1.3 Adaboost -- Boosted Cascade of Haar-like features

Proposed by Viola and Jones[14], Adaboost is an algorithm that has been applied for many face detection applications. The sliding window based algorithm constructs a strong classifier as a linear combination of weak classifiers (each contains a single filter) with the help of Haar like filters [15].

3.1.3.1 Feature Extraction

Figure 3.1.3.1 (left) lists some of the Haar filters that are adapted by Adaboost. Applying a template on the face image as in *Figure 3.1.3.1 (right)*, the value of this feature will be the sum of the pixel intensities in the white section over that of the gray section. These filters can be scaled to search for features over the sub-windows of the image.

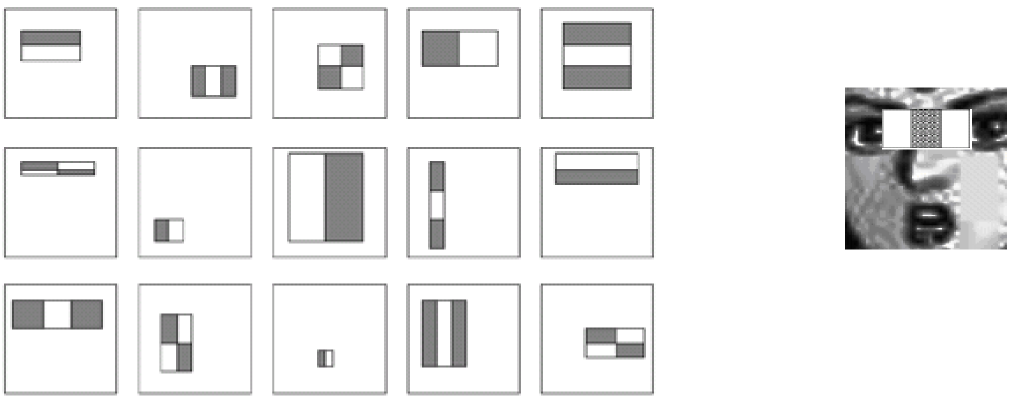


Figure 3.1.3.1 *(left) Haar features adapted by Adaboost; (right) Applying feature on image [26]*

3.1.3.2 Training

Once the feature to be used is defined, Adaboost then move onto the job of building a strong classifier from training the weak classifiers (*Figure 3.1.3.2a*). Within a sliding window, only a small portion of the features are needed to form a strong classifier. Given some sample images (x_1, y_1), …, (x_m, y_m) [y=1 for positive image, otherwise y=0], the strong classifier is created as follows [26][27]:

1. Initialize weights $D_1(i) = 1/m$
2. For t= 1 to T (number of weak classifiers)
   - Normalize the weights

- For each filter *j*, train a classifier h*j*, which is limited in a single filter; where the error *e* is $\sum D_t(i)\,|h_j(x_i)-y_i|$
- Find the best weak classifier that is of minimum error *e* with respect to the distribution D_t. (so that there is less error)
- Update the weight by

$$D_{t+1}(i) = D_t(i)\,\beta_t \exp(1-|h_i(x_i)-y_i|) \text{ where } \beta_t = e_t/(1-e_t)$$

3. The output strong classifier is

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \quad \text{where } \alpha_t = -\log \beta_t$$

Table 3.1.3.2 *Training of Adaboost Classifier*

When T weak classifiers are determined they contribute in a weighted vote for the final strong classifier; thus as mentioned earlier, the strong classifier is built from a linear combination of weak classifiers. *Figure 3.1.3.2a* is a diagrammatic view of the training process of the construction of the strong classifier and *Figure 3.1.3.2b* gives an example of how the for-loop in step 2 is done. It can be observed that the earlier the stage in the loop, the less number of weak classifiers are selected, the detection rate is better and tends more to 100% of detection rate. However if a small number of weak classifiers are chosen, the false detection rate will also increase; therefore this is a tradeoff, so for accuracy, many cascaded classifiers should be selected. Thus during the training stage, there are few concerns: if a fast cascade is required, less weak classifiers are selected, making the training process faster and more prone to 100% of detection rate but the classifier is not that “strong” provided that it includes only a few weak classifiers and a numerous of false detection is expected. Another concern is how to determine the number of weak classifiers are needed in producing a detection result which minimize the reduction in false positives (false detection) and maximizing the decrease of true positive. To deal with these concerns, each stage should be trained and the result is estimated, then the next weak classifier is added onto the cascade and trained again. The process stops when it produces the best result. But this process is very time-consuming.

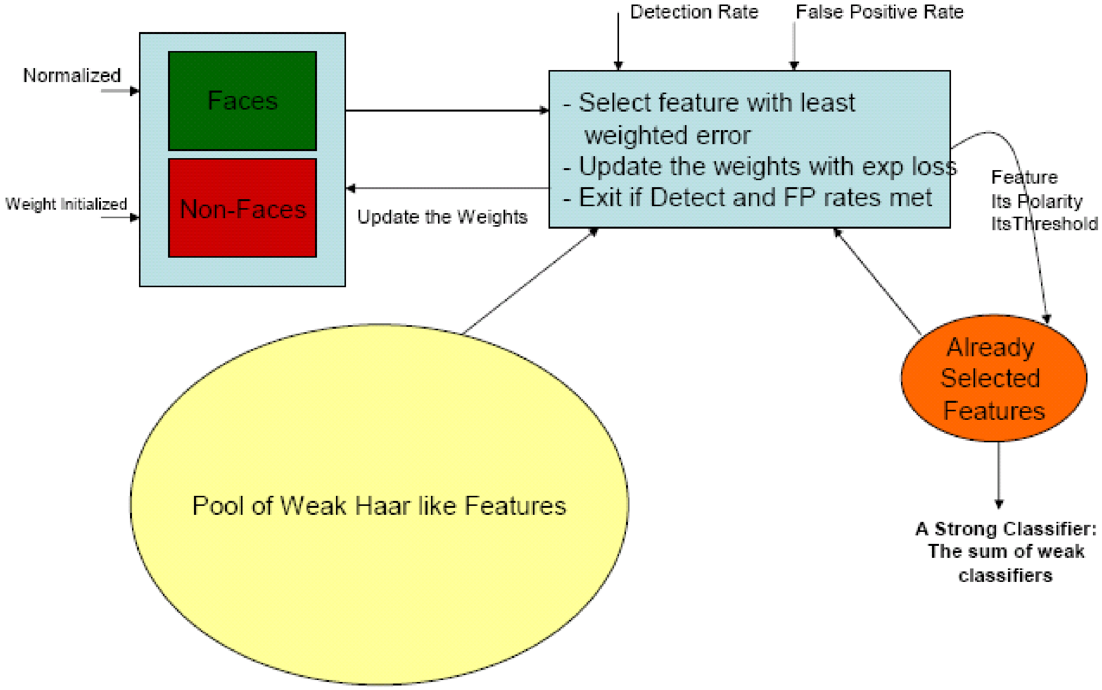


Figure 3.1.3.2a *Diagrammatic View of Adaboost Training[26]*

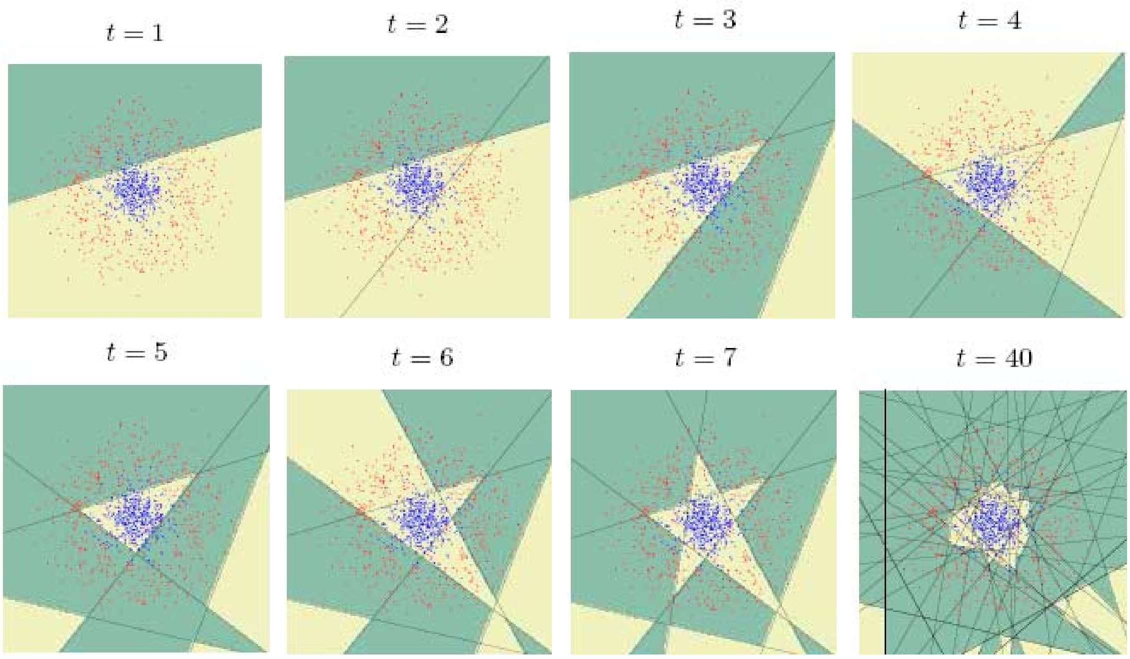


Figure 3.1.3.2b *Classification results for applying different number of weak classifiers[27]*

3.1.3.3 Detection

Once the strong classifier is obtained, we can proceed to the detection phrase. The concept here is similar to that of the training stage, by which the first classifier should return most faces, and the second will cut off more false detected objects (as shown in *Figure 3.1.3.3*), etc.

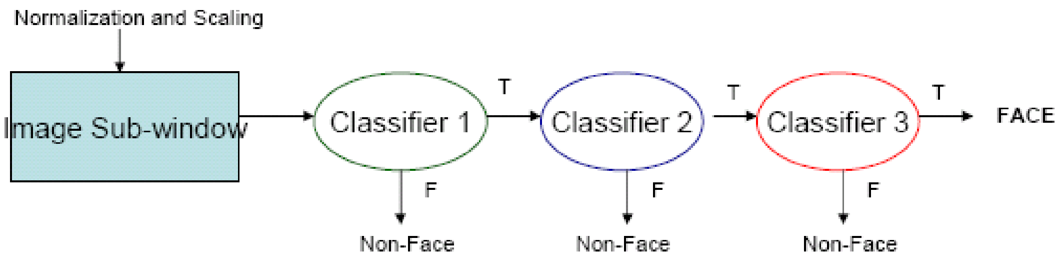


Figure 3.1.3.3 *Detection by using a cascade of weak classifiers to form a strong classifier [26]*

3.1.3.4 Detection on Comic Characters

In this project, a 21-stage Adaboost strong classifier is used to detect faces in a given image. Although it takes quite a while in training, the detection part is speedy. This is an advantage of Adaboost.

3.1.3.5 Adaboost on face recognition

Face recognition can also be done by Adaboost[18] where the positive images of a character class and the negative images are not of the character class. But to provide a good classifier, a large number of sample images have to be obtained, with at least 1000 positive images and 5000 negative images in addition to exhaustive training for minimum 2 weeks can give us a classifier for 1 single class. In this project we simply do not have such kind of resources on the characters image of 1000 per class.

Moreover, the performance of Adaboost on face recognition is not too good when there are a large number of classes involved.

3.1.4 Neural Network

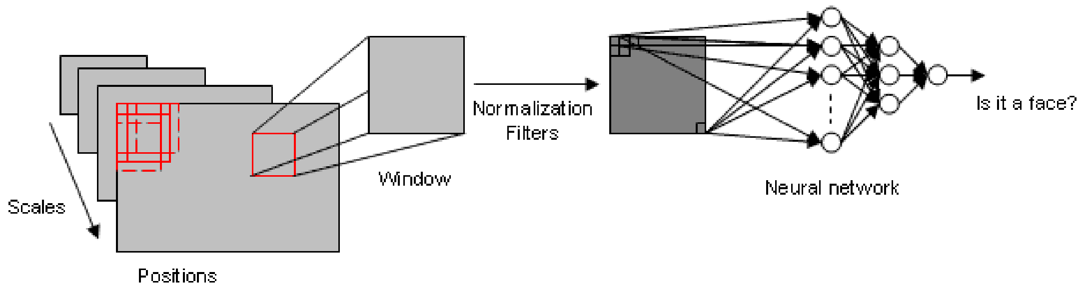


Figure 3.1.4 *Neural Network diagrammatic overview[28]*

Similar as Adaboost, Neural Network, coined by Rowley et al [5], works by sliding windows. An input comic image is to be scanned by sliding windows of different scales, in which these windows will be fitted in to a neural network. Having trained how to recognize a face, the neural network would be able to determine whether the input window contains a face. The Neural Network Library being distributed under GNU General Public Licence is acquired to demonstrate the experiments in Chapter 5 [28].

3.2 Face Recognition

The roadmap of face recognition techniques to be discussed is shown on *Figure 3.2*, where PCA and LDA will undergo Subspace Training and Subspace Project; Bayesian and EBGM will train and test on a different path.

Along with the face imageries, the coordinates of eyes of each face images are assumed to be obtained before the normalization is operated. All the algorithms are provided by Colorado State University (CSU) Face Identification Evaluation System (version 5.0) [19].

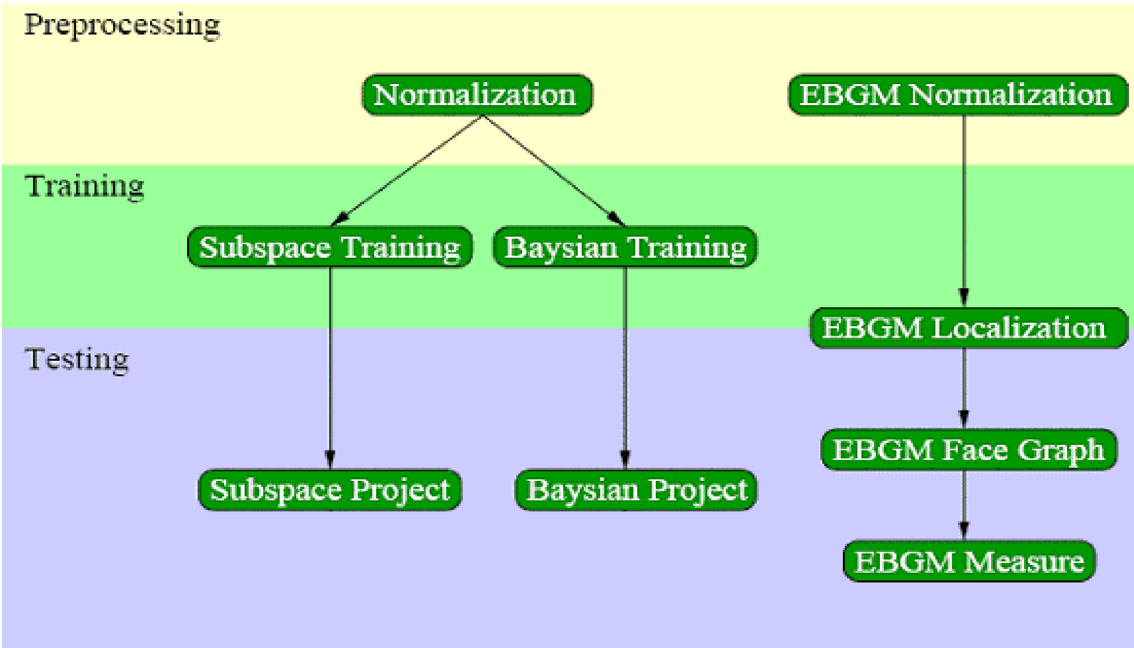


Figure 3.2 *Roadmap of PCA, LDA, Bayesian and EBGM (modified diagram from [19])*

3.2.1 Preprocessing

Normalizing the images before applying onto the training process is a crucial step in classification and the schedule is adopted from [19]. The imageries obtained first have to be transformed to gray scale images, which in turn to be normalized into imageries that are portable for the training or testing stages of different algorithms.

Procedures for preprocessing:

1.	Resize the image to 130 x 150 x 8BPP
2.	The gray value of the gray scale images is cast into decimal
3.	The image will be rotated such that the two eye points will be lying on the same y coordinate.
4.	The redundant part which is supposed not to be carrying any face feature will be cropped by an ellipse mask.
5.	Normalize the histogram of the image.
6.	Normalize the pixel values such that mean and SD is equal to 0 and 1 respectively.

Table 3.2.1 *Procedures of Preprocessing*

Figure 3.2.1 *Normalized image of a comic face*

3.2.2 PCA – Principle Component Analysis

Principle Component Analysis (PCA) [20], also named as Karhunen-Loeve transform in functional space, is widely used to reduce dimension. Under face recognition PCA is going to find the most accurate data representation, that is the maximum variance, in a lower dimension space and perform a similarity measure between the given data.

3.2.2.1 Training

So during training stage the eigenvectors best represents the input data are found. For instance, in *Figure 3.2.2.1*, the diagram on the left side is not an ideal projection of maximum variance as it exhibits large projection error; an optimum maximum variance is shown on the right diagram.

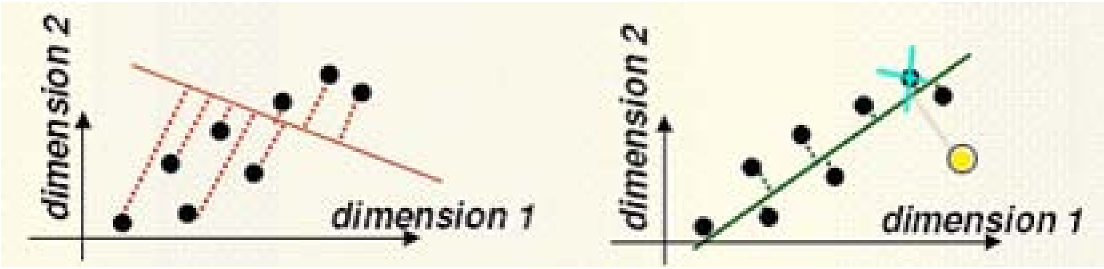


Figure 3.2.2.1 *Determination of the maximum variance by PCA (modified from [29])*

Given an image, it can be represented by a vector of pixels, in which the attributes of the vector is filled in by the grayscale value of the respective pixel. For our example, a m by n image can be represented by a 1 by mn vector. Then the image is said to be located in the mn dimensional space, where this is the original space where the image will be located at. Then the procedures are lists as follows [30]:

1. Given a set of N training images,

$$\{ x_1, x_2, ..., x_N\} \text{ in mn-space}$$

There are a set S with M number of faces, represent in vectors,

$$S =\{ x_1, x_2, ..., x_N\}$$

PCA will project it onto a d < mn space:

2. With these set of training images the mean image Ψ can be obtained, where

$$\Psi = (1/M)\Sigma x_v \text{ (where v= size of vector)}$$

3. Then the difference Φ between the input images and the mean image is defined by

$$\Phi_i= x_1-\Psi$$

4. Next we will find a set of M ortho-normal vectors (u_n) which best describes the distribution of data. In set of M, each attribute (k) is found by

$$\text{Max}[(1/M)\ \Sigma(u_k\Phi_v)^2] = \text{eigenvectors of k (for k = 1 to M)}$$

5. To find the covariance matrix Ω,

$$\Omega = (1/M)\ \Sigma\ (\Phi_v\Phi_v^T)=AA^T \text{ where } A = \{\Phi_1, \Phi_2, \Phi_3, \dots , \Phi_v \}$$

6. The eigenvectors can be obtained by,

$$\Omega V = \Lambda V \text{ (where } V \text{ is the set of eigenvectors associated with the eigenvalues } \Lambda)$$

Table 3.2.1 *Procedures of finding Eigenfaces*

As one may notice, PCA takes into account of every pixel intensity to be a feature and reduce the dimension of them to find the variance. Therefore under face recognition, it did not take into the advantage of known features such as eyes or nose points; also under PCA, no classification information is required to train the image.

For PCA that have been used for face recognition and gives an outstanding result, it is more likely that most of the faces are of registered image, where the vector generated for all training and testing images will not have much discrepancy so the recognition job could be completed with less error. But rationally speaking, PCA will not perform that good under comic images.

3.2.2.2 Testing

In testing stage, exploiting eigenvectors from the training data, a similarity measure of the testing image with the data in the training stage can be measured by projecting the test image onto the face space, the closer the distance is, the more likely it will be of the same class. As illustrated in *Figure 3.2.2.1*, after the normal points (green) had undergone training, the maximum variance on the right is found. The subspace (green) for projection will be obtained, and given a test data (the yellow circled point), it will be projected onto the subspace and the distance between test data projected point and other training data on the projection can be measured, apparently, the closest point with the test data is the normal data which is marked by the light blue cross, thus PCA will say this data should be a class of the yellow test point and will rank it on the first place in the recognition result. The distances could be measured by various kinds of distance measures.

3.2.3 LDA – Linear Discriminant Analysis

PCA works on the face space by simply entering the whole set face images instead of considering the entered face image is of which class during training. But the direction of maximum variance determined by PCA might not be that useful in classification as a good representation of the data (maximum variance) does not imply that it will be useful to the classification of data. *Figure 3.2.3a* illustrates an example when PCA classification cannot separate the classes. Logically, by taking the advantage of known image classes, LDA[21], which aims on finding the best subspace so that the data can be well separated as classes of objects, may be obliging to accomplish the identification job.

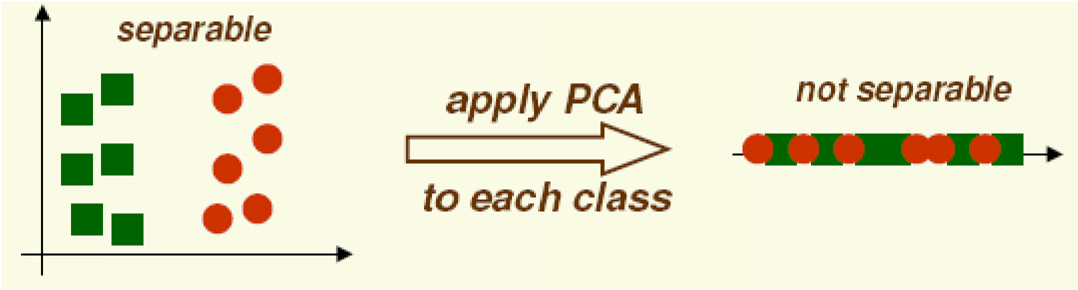


Figure 3.2.3a *Problem with PCA in classification[31]*

Figure 3.2.3b explains how Fisher Linear Discriminant (FLD) is able to separate two classes in 2D dimension. On the left diagram, the separation plane, lies between 2 classes, has bad result on classification as the projection of the two classes are mixed; where the diagram on the right, the projection of the classes onto the blue plane can be well separated. LDA tries to find a linear transformations which is similar to the case on the right size, which maximize the within class scatter and minimize the between class distance.

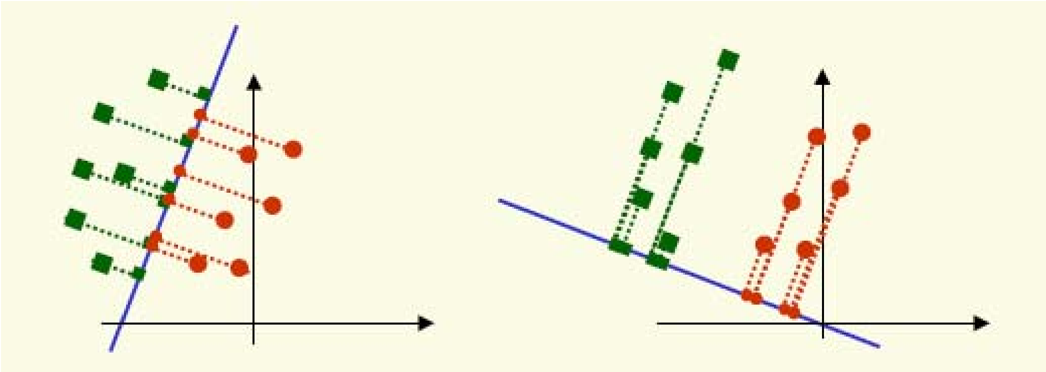


Figure 3.2.3b *FLD tries to find a projection that can maximize the between-class distance [31]*

3.2.3.1 Training

LDA is trained by applying PCA to reduce the dimensionality of the feature vectors, thus by PCA the maximum variance of the training data is found, and then LDA will further reduce the dimensionality meanwhile maintaining the class distinguishing features. Thus here LDA can be described as a combination of PCA and LDA.

3.2.3.2 Testing

The testing part is just the same as PCA but using the trained subspace in LDA.

3.2.4 Bayesian Intrapersonal/Extrapersonal Classifier

Two of the mentioned recognition algorithms project face images onto a subspace by taking assumption that the projection of the face images onto the subspace will have a tighter cluster of points, if they belong to the same class. Instead of representing the imagery as points on the face subspace, the spanned space of the difference between two face images are to be considered by this classifier, which are the intrapersonal (same character) and extrapersonal (different character) subspace. Moghaddam and Pentland [22] propose that the intrapersonal and intrapersonal from different classes could be represented by Gaussian distribution [23].

3.2.4.1 Training

The density estimation is done by PCA, training the classifier for two times: first for the set of images of intrapersonal difference and second for extrapersonal difference. This is done as to defining the distribution of Gaussian.

3.2.4.2 Testing

Matching is done by computing the possibility of the differences of testing and trained images to see if they are from the intrapersonal or extrapersonal space. By projecting the probe image onto each space, the probability of where the probe image is come from is computed.

3.2.5 EBGM—Elastic Bunch Graph Matching

Contrived by Wiskott et al. [10], EBGM utilizes the fundamental nature of human face and extract the features of those fiducial points to differentiate from class to class. As mentioned from the roadmap in *Figure 3.2*, it undergoes a totally different classification process from the other recognition methods mentioned in previous sections. EBGM have its own preprocessing, then training is done by EBGMLocalization and after obtaining the face graphs of the face images, distance measure can be finally computed. In this project, the CSU EBGM, which is based on the thesis of Bolme from Colorado State University [24], will be applied.

3.2.5.1 Normalization

To enhance the localization performance, EBGM will exploit another normalization process owing to the algorithm's specialty. As EBGM took into account of the head of the imagery but not only to the face, more features will be included in the preprocessing outcome compared to the preprocessing described in 3.2.1, for which the top of the head of the latter is occluded after preprocessing. The image on the right of *Figure 3.2.5.1* are the original image of the left, which is the normalized face image processed by EBGM, note that it comprises of more features than *Figure 3.2.1*. The EBGM normalized face images will be of 128 x 128 x 8BPP.

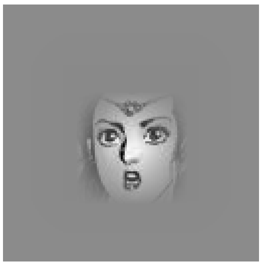


Figure 3.2.5.1 *(left) Image output after undergoing preprocessing of EBGM;*

(right) original cropped image

3.2.5.2 Landmark Localization

Going though this process the algorithm can locate the feature locations on the set of preprocessed training images, and hopefully a bunch graph can be generated. Before automatic landmark localization of the preprocessed images is proceeded, the landmarks of the training imagery have to be selected manually. The 25 landmarks are listed in *Figure 3.2.5.2*.

1. LEye	6. LEyeBrowInside	11. CNoseBottom	16. LMouthCorner	21. LFaceEdge
2. REye	7. REyeBrowInside	12. LNoseBottom	17. RMouthCorner	22. RFaceEdge
3. CNoseBridge	8. LEyeBrowOutside	13. RNoseBottom	18. CTopHead	23. CChin
4. LEyeBrowPeak	9. REyeBrowOutside	14. CMouthTop	19. LTopHead	24. LJaw
5. REyeBrowPeak	10. CNoseTip	15. CMouthBottom	20. RTopHead	25. RJaw

(L = Left, C = Center, R = Right)

Figure 3.2.5.2a *The 25 landmark features that have to be known for the construction of a model graph[24]*

After locating all the landmarks, they have to be connected together to form a model graph, which is similar to *Figure 3.2.5.2b*. Then, the algorithm will load all the model graphs and extract the corresponding Gabor wavelets from the image to serve as the feature and add them onto the respective jet in the bunch graph. For example if we have 6 model graphs, all the REye jet from the 6 model graphs will be extracted and be appended onto the face bunch graph, *Figure 3.2.5.2c* illustrates this example with 9 landmark jets.

Figure 3.2.5.2b *(left): model graph on a real person image from [24]*

(right): model graph with landmarks on the preprocessed image;

the crosses (left) and dots (right) in red represents the landmark jets;

where the lines(blue) denotes the connection of interpolated jets

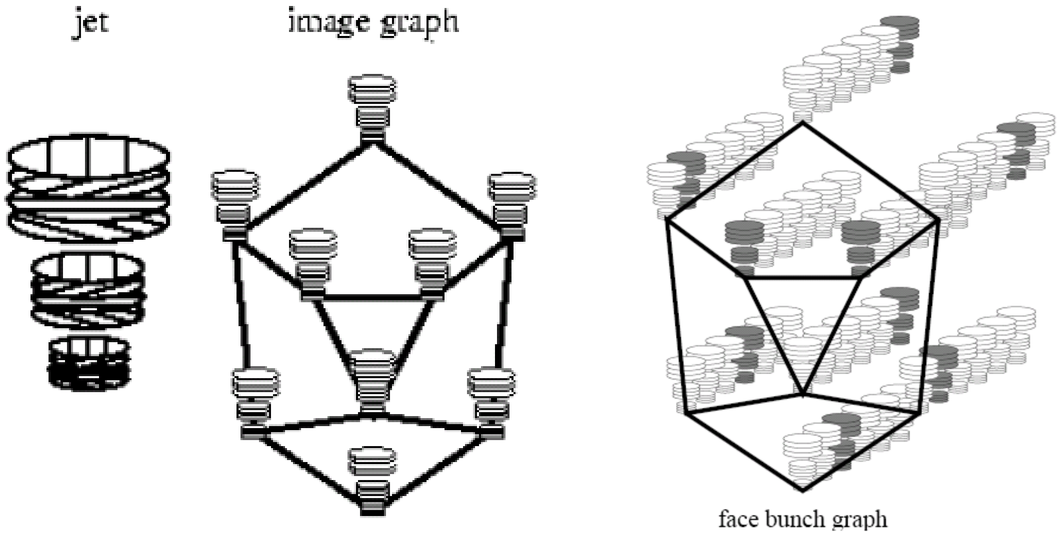


Figure 3.2.5.2c *left: a jet; center: image graph with 9 landmark jets;*

right: face bunch graph[10]

3.2.5.3 Face Graph

To be able to test all the images in the database, graph descriptions for the entire images have to be constructed. This is done similarly as above with the aid of the bunch graph created in the previous step. For the landmark location of every test image, they can be estimated by the known position of eye coordinates, for example the coordinates of CNoseBridge could be estimated as the coordinates lies between the eyes, in turn for other coordinates. Once all the automatic landmark localization are done, the image will be of no use to EBGM as the face graph will be the representation of the images. As a face graph file is much smaller than an image regarding to the file size, it is believed that the matching procedure is along more efficient.

3.2.5.4 Distance Measure

For recognition part, the probe face graph is to be compared to jets in the bunch graph to find a similarity measure. In the right most diagram of *Figure 3.2.5.2c*, the input face graph is to be compared with the jets on the corresponding jets on the bunch graph and the best fitting jet in each of the bunch jets are selected accordingly, which is highlighted in grey. Afterwards, the average similarity of the Gabor jets are computed between the testing data and each of the best fitting jet in the bunch graph. The smaller the distance is, the more likely the test data is of a class of that training data.

3.2.5.5 EBGM on Comic Images

Since the eye points are already a known feature, the rest of the points can roughly be estimated. As the progress of manually selecting the 25 landmark on the whole set of training images is exhaustive, those 25 points are roughly estimated in applying EBGM to the CBIR system developed.

Chapter 4 -- Comic Faces Image Retrieval System (MAIRE)

This section gives a detail description on the application that has been built under this project to cater for comic readers' needs; it is named as MAIRE (coMic fAces Image Retrieval systEm). The followings will focus on the functionally of MAIRE.

4.1 Overview

MAIRE is an executable implemented by MFC. With the aid of MAIRE, comic readers could be able to search a particular scene by specifying the character(s) that is related to the scene from a large set of comic images in the database. MAIRE performs its search by the face recognition.

4.2 System Structure

Figure 4.2a shows the use case of MAIRE and *Figure 4.2b* is the class diagram.

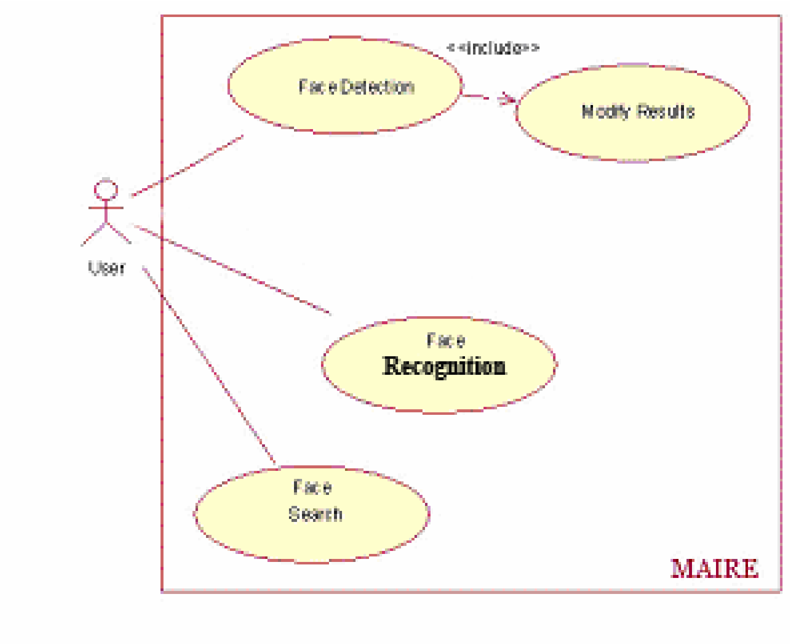


Figure 4.2a *Use Case of MAIRE*

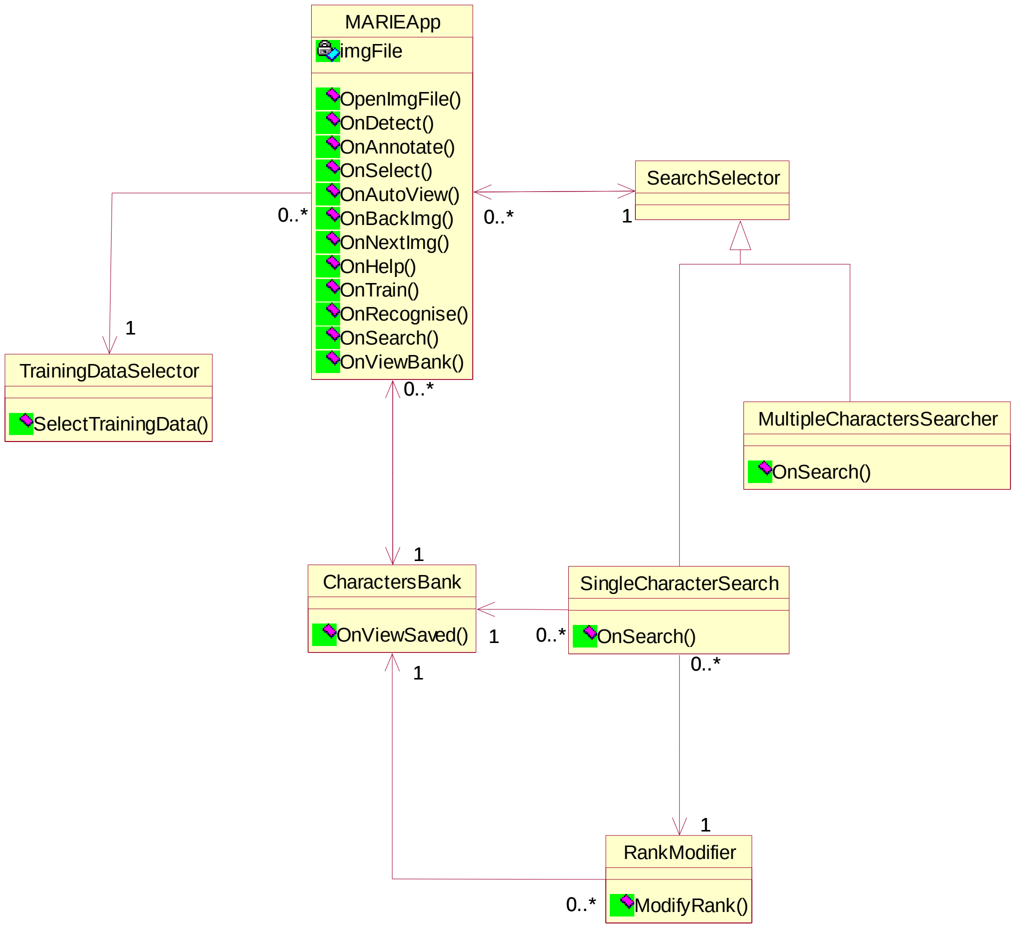


Figure 4.2b *the class diagram of MAIRE*[1]

[1] The classes that are insignificant to the system flow are not shown

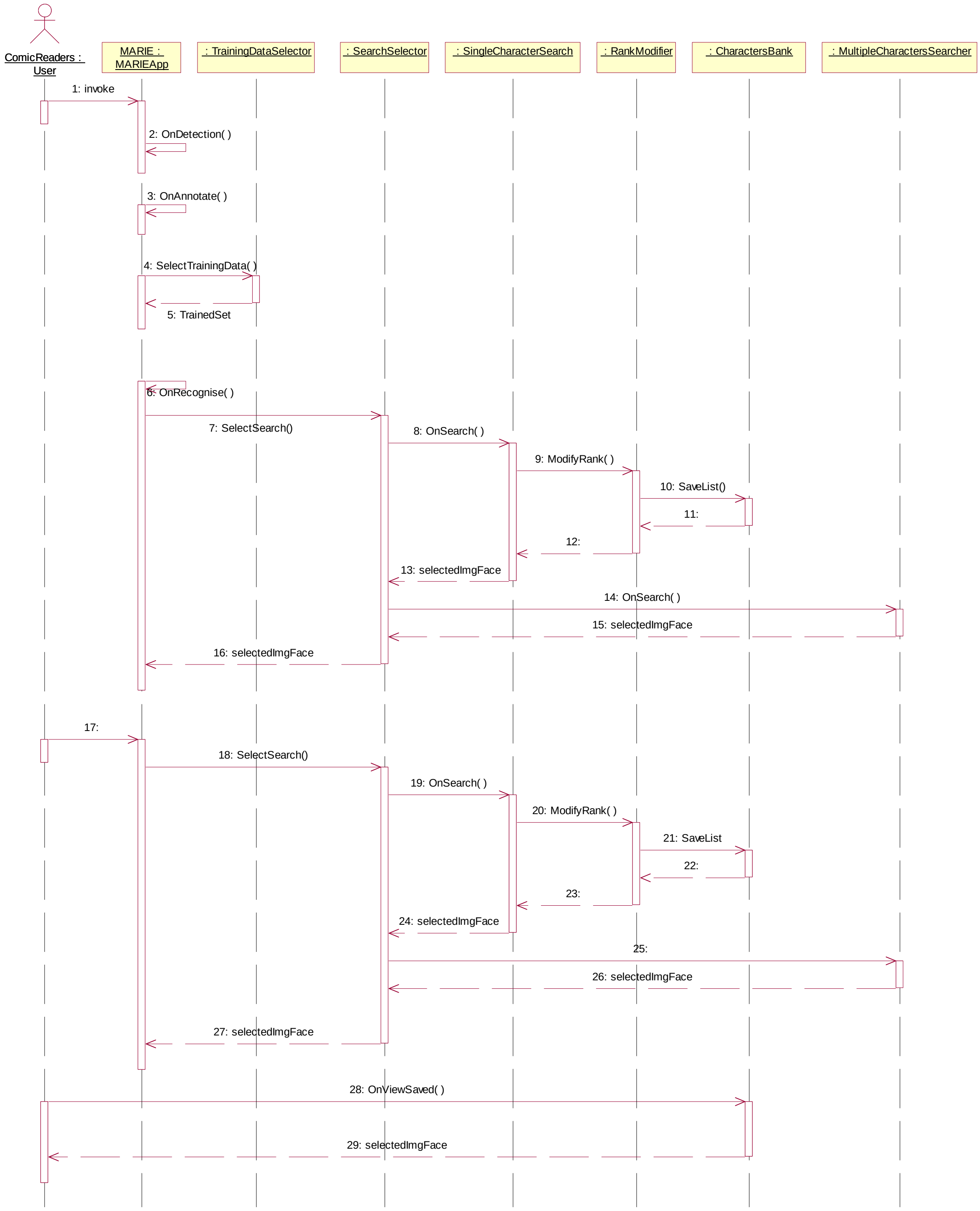


Figure 4.3a *The sequence diagram, stating the sequence flow of using MAIRE.*

4.3 Image Retrieval

In order to search for a particular scene, as specified by the sequence diagram (*Figure 4.3a*), users have to:

1. Specify the image folder of the desired comic set by opening an image that is located in the comic set. If face detection has been done before, the image will show the regions where the faces will be located by red bolded rectangles.
2. MAIRE provides 2 face detection techniques for users to automatically detect comic faces in comic images ---- the Adaboost and HSV skin color detection. Users can opt for either of them to perform face detection. The recommended one is Adaboost since it is faster and the localization of faces is more accurate. While MAIRE is working on finding faces, a progress bar will pop out to notify users the image MAIRE is working on. After MAIRE has found all the faces, the detected faces will then be displayed by red bolded rectangles and the corresponding eyes are marked by 2 eclipses.
3. As the detection performed by the system is not perfect, some amendments of the results are suggested before proceeding to the process of recognition. By using the rectangle tool and the eye tool, users can add in undetected faces, delete or modify the localization of faces and eyes. For convenient use, users can traverse the images back and forward by the back and forward buttons. Once they moved from image to image, the amendments, which have been done by the user on the former image, will be saved automatically.
4. After the annotation of faces has been done, MAIRE is ready for face recognition. MAIRE has 4 different face recognition techniques for users to select according to their preferences, in which includes PCA, LDA, Bayesian intrapersonal/ extrapersonal classifier and EBGM. As EBGM outperforms the other algorithms, it

is advised to be used in the recognition part. To be able to recognize, training has to be done first; by clicking on the training button of the selected face recognition techniques, the preprocessing of all the annotated faces in the image set will be done and a dialog will be popped out for users to specify the data for training. After specifying the faces for training, MAIRE will then perform training on those faces. It may take a while for the whole training process to be done, depending on the number of training images and the face recognition technique.

5. MAIRE is prepared for recognition after training has been completed. To operate recognition, user should have clicked on the recognition button of the trained algorithm. By then, MAIRE will perform the similarity distance measure of the test images, in which the test images can be obtained by the whole set of comic annotated faces excluding the training faces.
6. By the time the recognition process is finished, the user can perform query and searching. MAIRE will ask whether the user want to search for a single character or multiple characters. Then the dialogue of user's choice will be instantiated for query. Once the desired face image is found, MAIRE will be able to locate the comic page that the face image origins and the user will then be able to find that particular scene.

4.4 User Interface

To cater for different types of comic readers, MAIRE provides a graphical user interface for viewing and searching comic images. MAIRE is designed to be as similar as the common window system so that MAIRE starters will have a familiar feeling on MAIRE. The major functions in the toolbar are specified in alphabetical orders in *Figure 4.4a*.

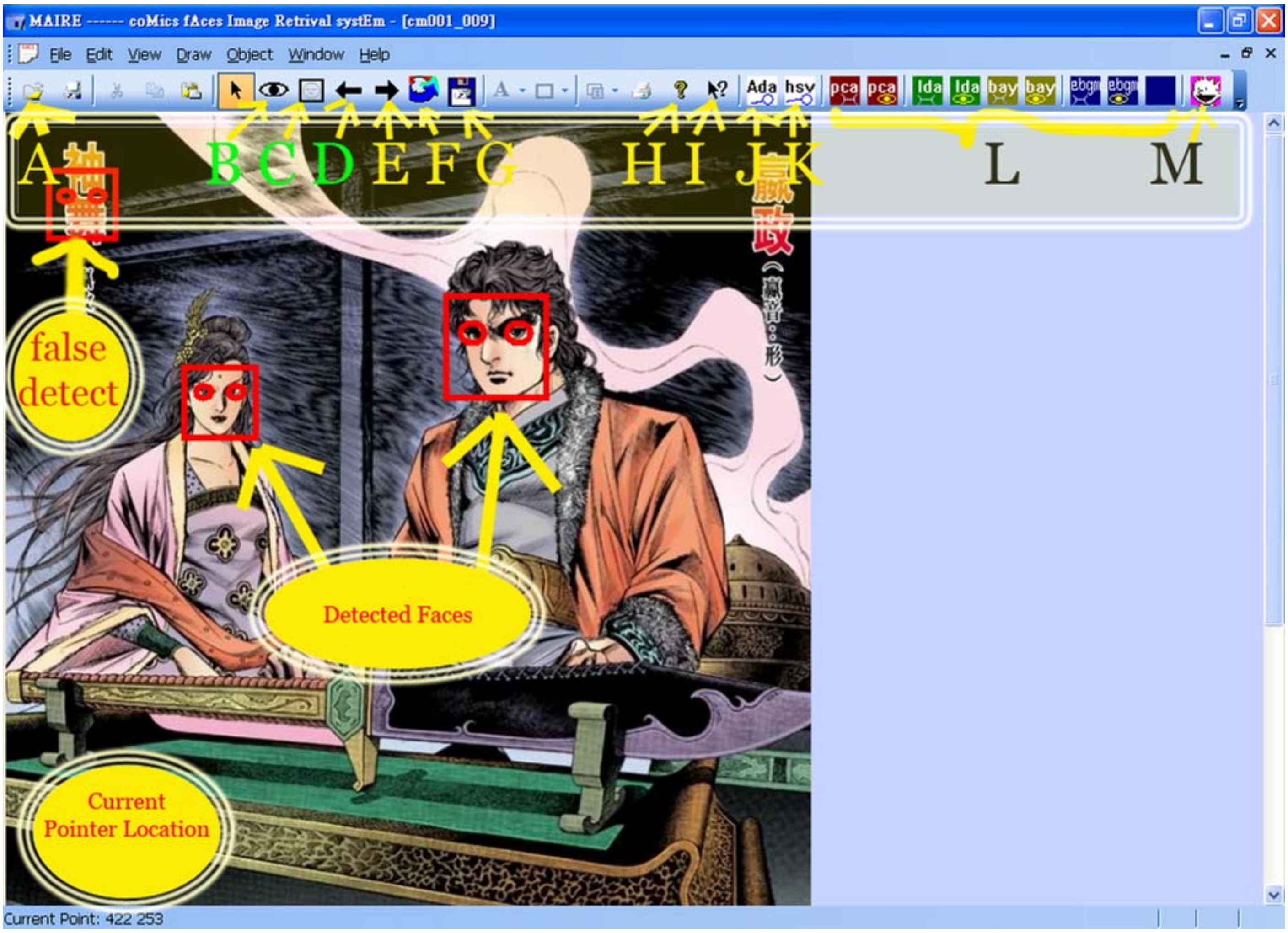


Figure 4.4a *User Interface*

Major functions:

A: Open a comic image page

B: Eye tool for marking the eyes of face image

C: Rectangle tool for locating face regions

D: Traverse previous image page

E: Traverse next image page

F: Automatic viewing of comic image pages (i.e. MAIRE will show the next comic page automatically after 6 seconds)

G: Retrieve saved characters face images (Character Bank)

H: User Manual Online Help

I: MAIRE Application Detail

J: AdaBoost Detection

K: HSV Skin Color Detection

L: Training and Recognition for the 4 face recognition techniques

M: Search comic character (can only be activated after training and recognition has completed for at least once)

Minor functions in annotation of face and eye region:

Select tool, to select the annotated object such as rectangles or eclipses.

Change the color of an object (rectangle or eclipses). The default color is red.

Change the line width of object.

4.4.1 Performing Detection

Figure 4.4.1 shows the progress of AdaBoost detecting faces in the comic images in a progress bar.

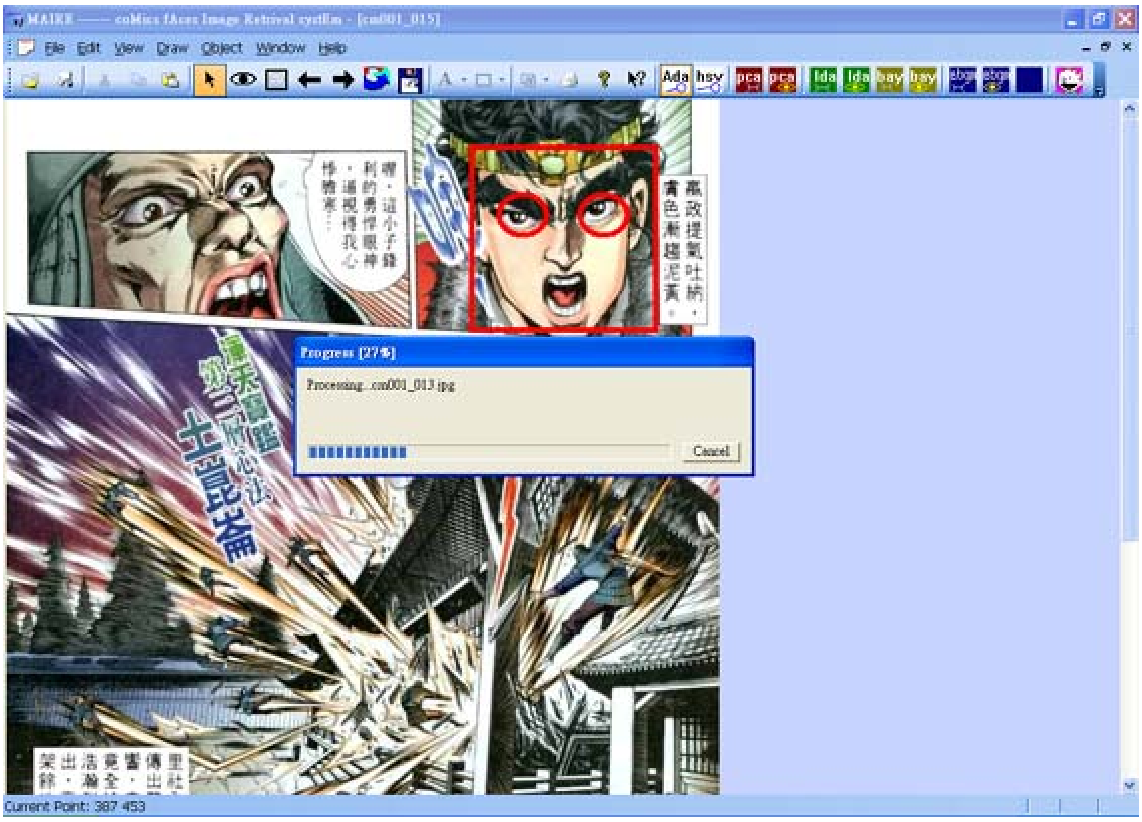


Figure 4.4.1 *Progress Bar*

4.4.2 Training Data Selector

In the training mode, once MAIRE has collected all the image faces from the comic set, a menu will come up for users to specify the class(es) he wants to train on (*Figure 4.4.2*). The maximum number of classes MAIRE can handle is 10000. The left panel displays the added face images by the user that are of the same class; while users can cull the face image in the database generated in the detection process on the right panel. To add a training image to a class, what the users have to do is simply click that particular face image and click “add to train” button. Users can create a new class of characters by just pressing the “Create New Class of Character” button. Upon entering the training set name and click "OK", MAIRE will then perform training on those set of characters.

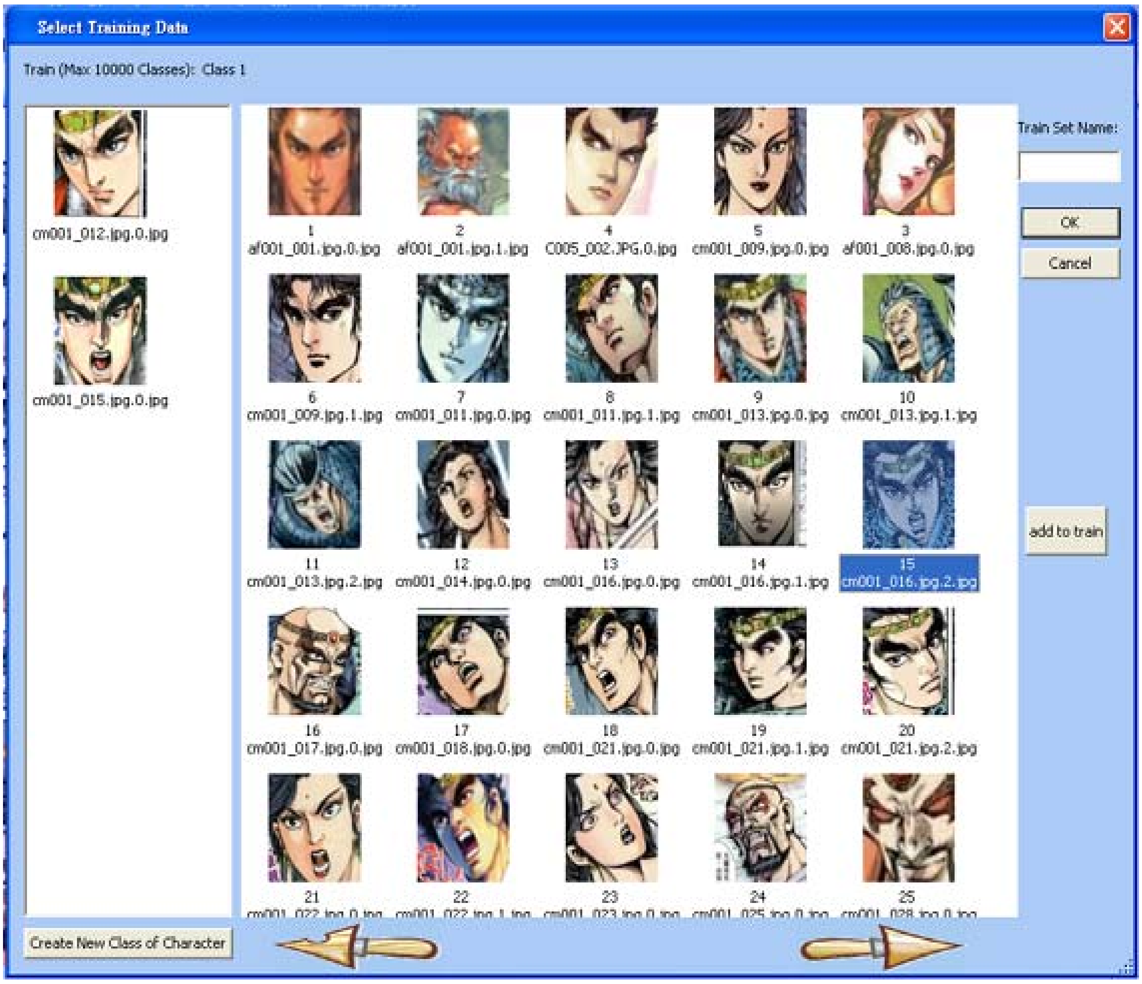


Figure 4.4.2 *Training Data Selector*

4.4.3 Search Selector

When the recognition part is completed, users can choose to perform a search on single character or multiple characters (*Figure 4.4.3*).

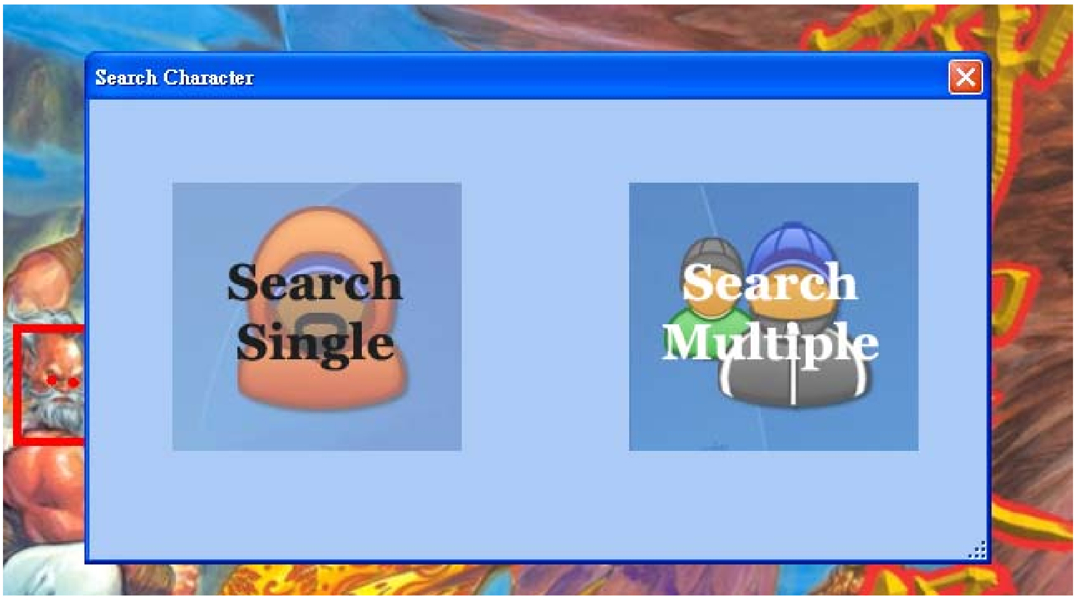


Figure 4.4.3 *Search Selector*

4.4.4 Single Character Searcher

If the user specifies searching for single character, he first picks the character he wants to search by traversing through the face image database (*Figure 4.4.4*). When he finds the desired character, clicking on it and press "Search Character" can then make a query for that comic character.

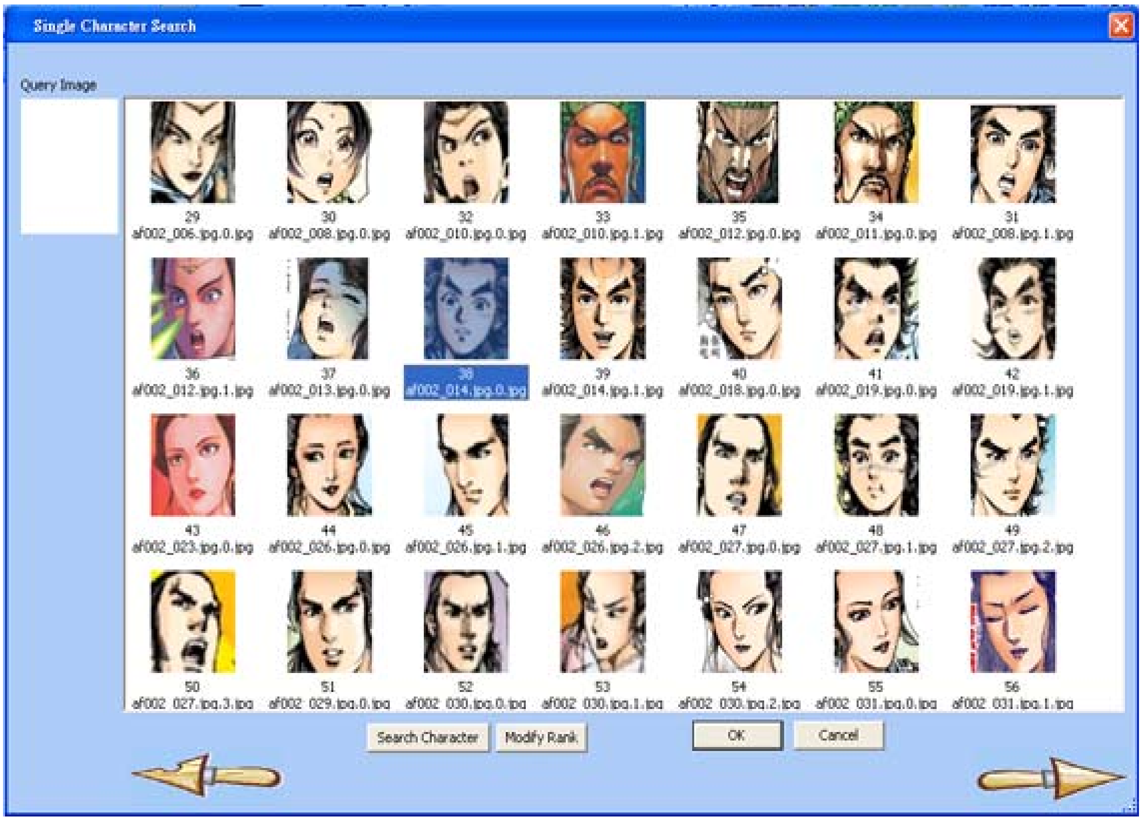


Figure 4.4.4a *Single Character Searcher*

MAIRE will then return the list of images which are more likely to be the query character in top ranks (*Figure 4.4.4b*). The query image is shown on the top left corner. The first page lists 28 rankings, and to view other rankings, user can click on the "next" arrow button; it is believed that the lower the ranking, the chance of finding the desired character will be lower. The rank of each image is shown under the thumbnail of face image. After viewing the results, if the user still wants to perform another searching, he can barely choose the face image and search for that character again; where, if the user already found the desired face and would like to read in detail what is going on with that particular scene, clicking "OK" will bring the user to that particular image page on the main application. However, if the user is not satisfied with the result that is given by MAIRE, he can modify the ranking of the query character by the "Modify Rank" button.

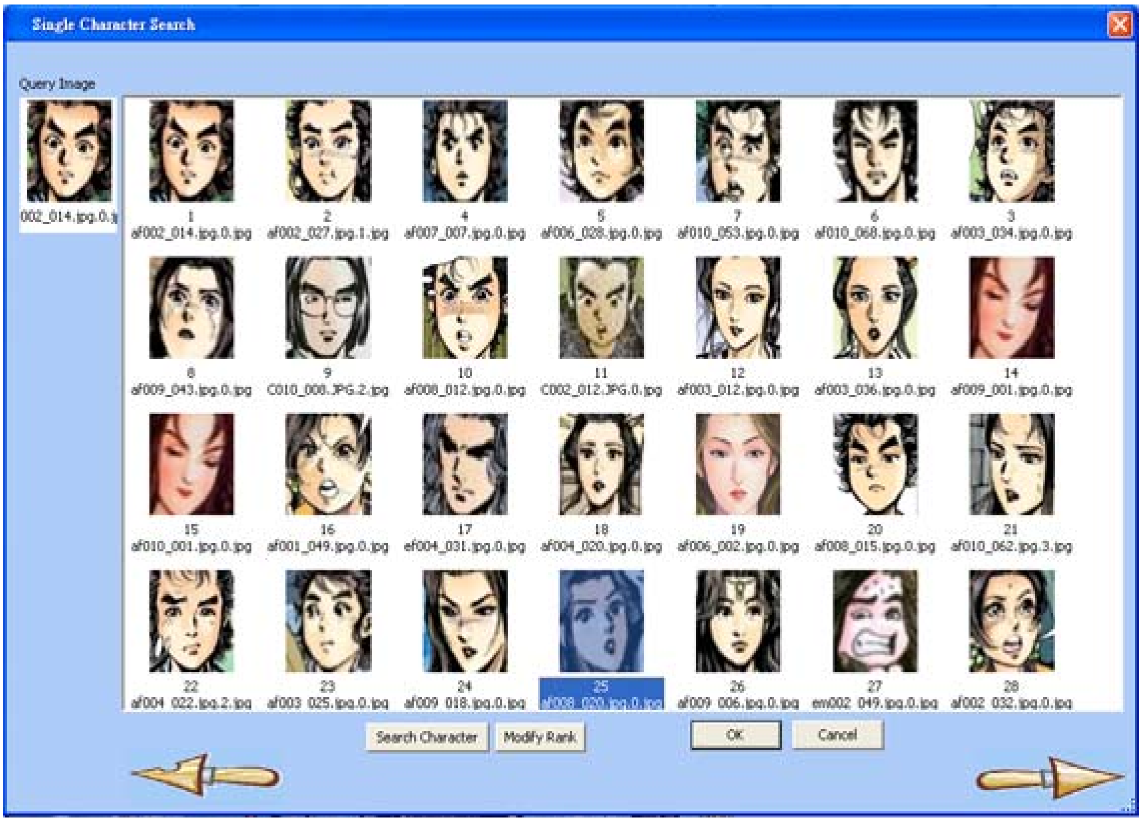


Figure 4.4.4b *The result of Single Character Searcher after a query is made*

4.4.5 Rank Modifier

The operation of rank modifier is similar to the procedure of specifying training data. Here, the query image of previous search is also shown on the top left corner of the rank modifier (*Figure 4.4.5*). The right panel displays the ranking that the user wants to modify in the previous search result. To add the images on the save list, select the face image from the rank panel and add the face image. Once the list of the query character has been completed, the user can give this query character a name and so the list can be saved into the Character Bank.

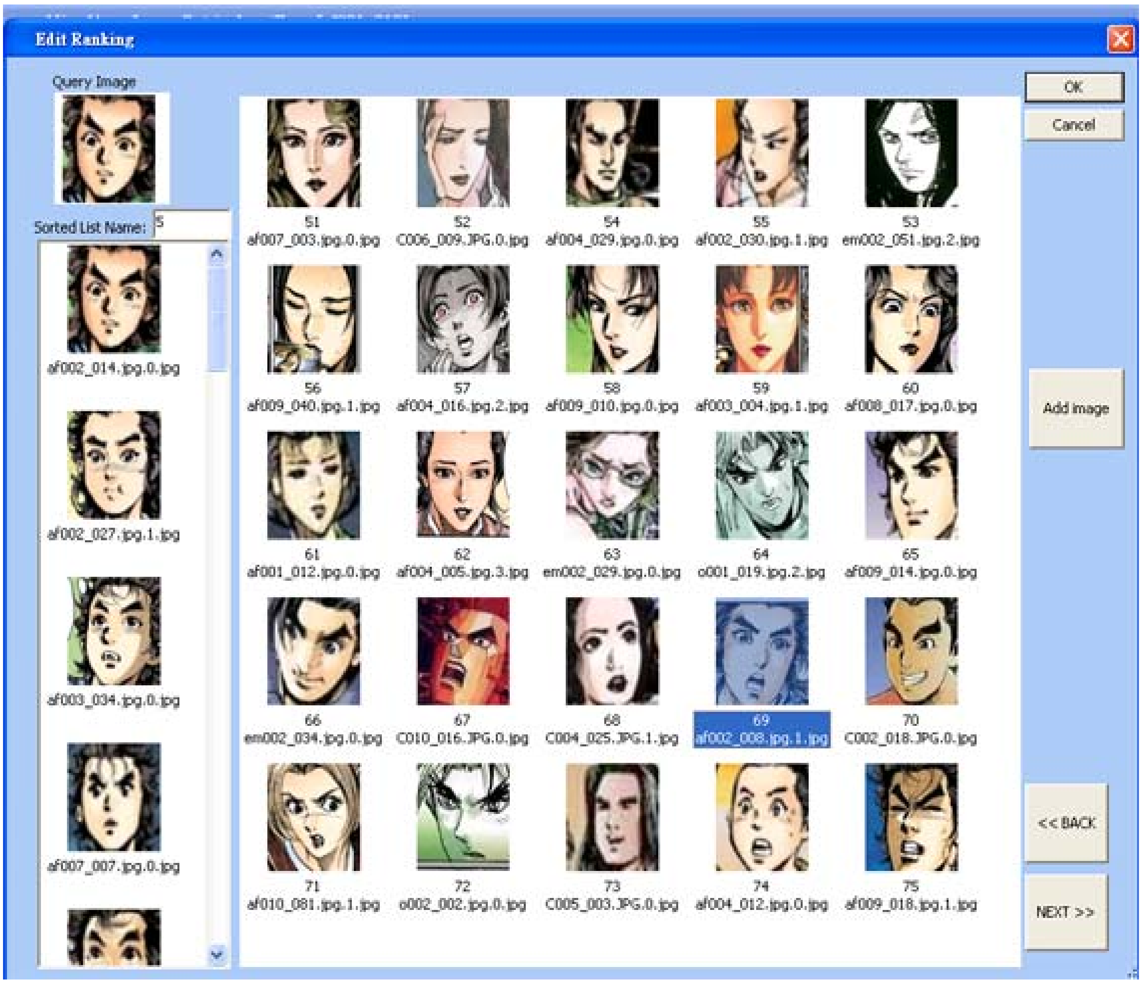


Figure 4.4.5 *Rank Modifier*

4.4.6 Characters Bank

Once the rank list of the characters is saved, the list can be retrieved by the users at anytime by the Characters Bank. The list of saved characters can be selected by the drop box on the top of the dialogue (*Figure 4.4.6*). If the user noticed that the desired face image is on the list, clicking on it will return the image page where the face image origins.

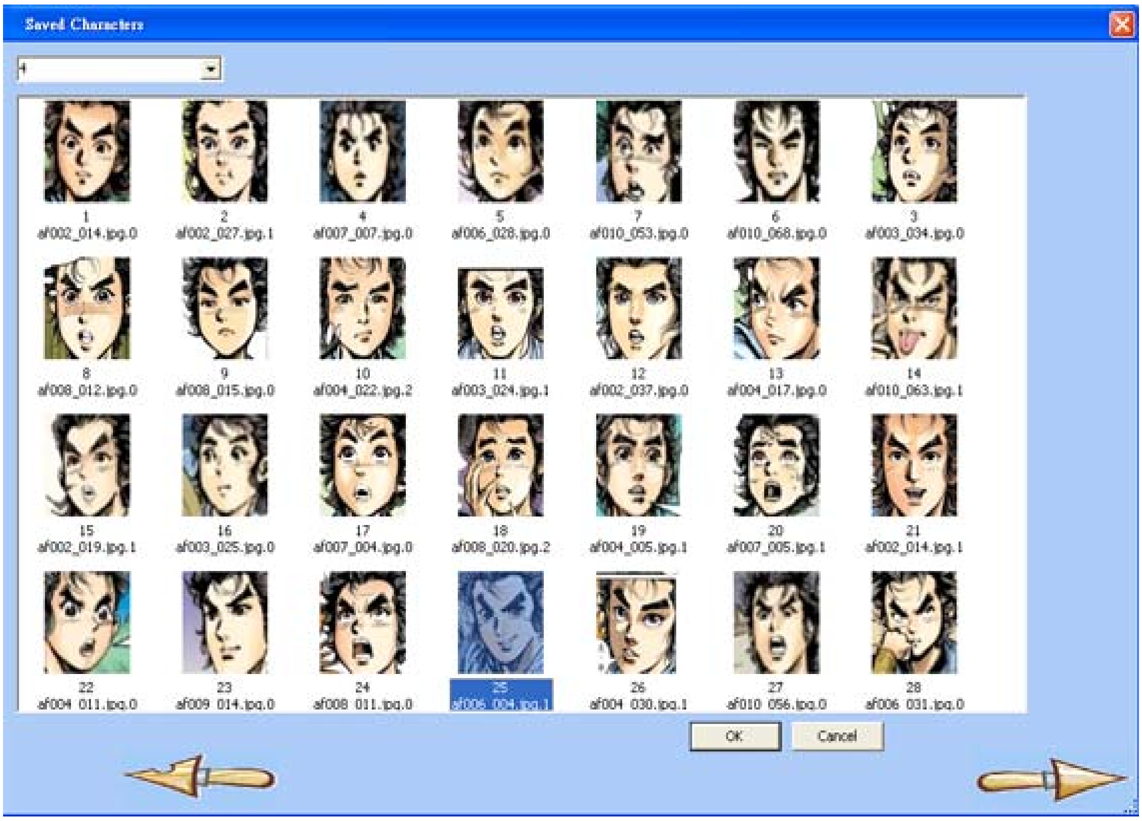


Figure 4.4.6 *Characters Bank*

4.4.7 Improvement on the query result

Once the comic character is saved in the characters bank, they will be shown as top rank on performing a new search. An example is shown on *Figure 4.4.7a*.

Making a query of the same character from the saved list, if the query face image is saved as a record in the characters bank, (not necessarily the same query face image as before), the single character searcher will retrieve the saved list from the bank and rank them on top of the query result, which in turn increase its performance, *Figure 4.4.7b* shows the top rank result of performing a search on the query result without saving anything in the characters bank; where *Figure 4.4.7a* is the result of saving the list in *Figure 4.4.6*.

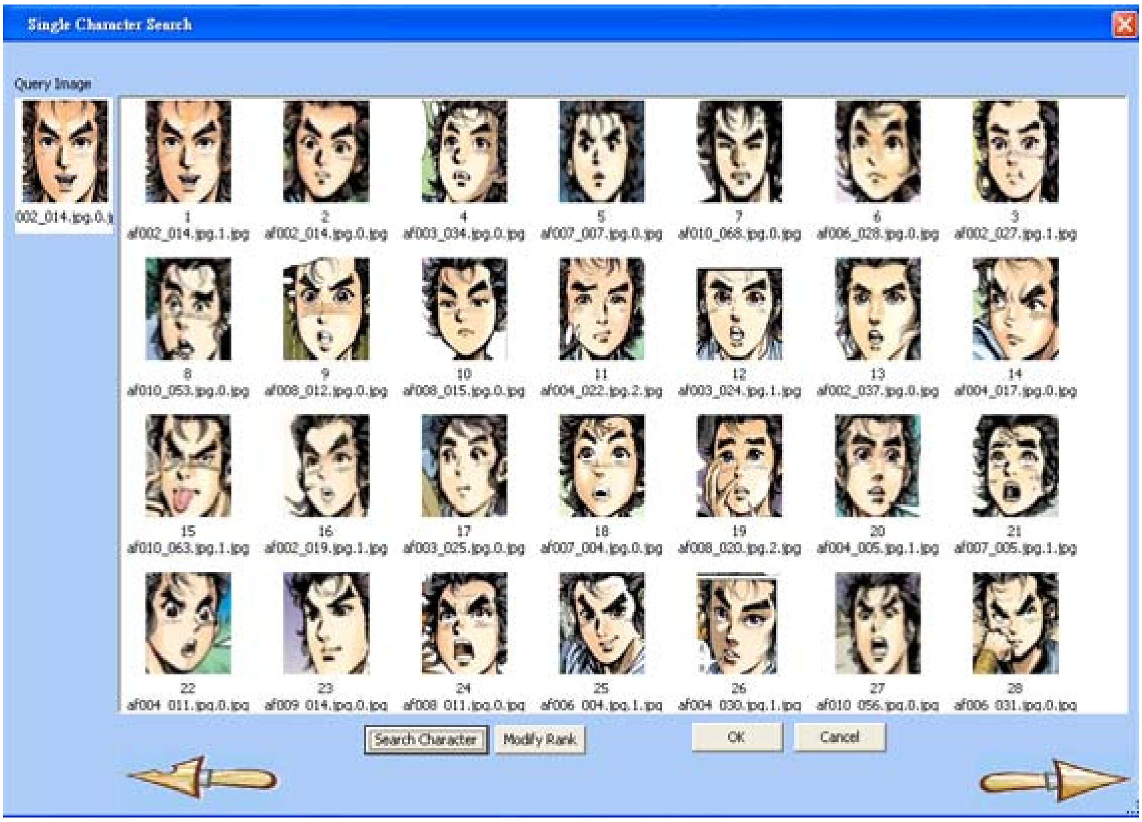


Figure 4.4.7a *The query result of a character who had its face image saved in the Bank*

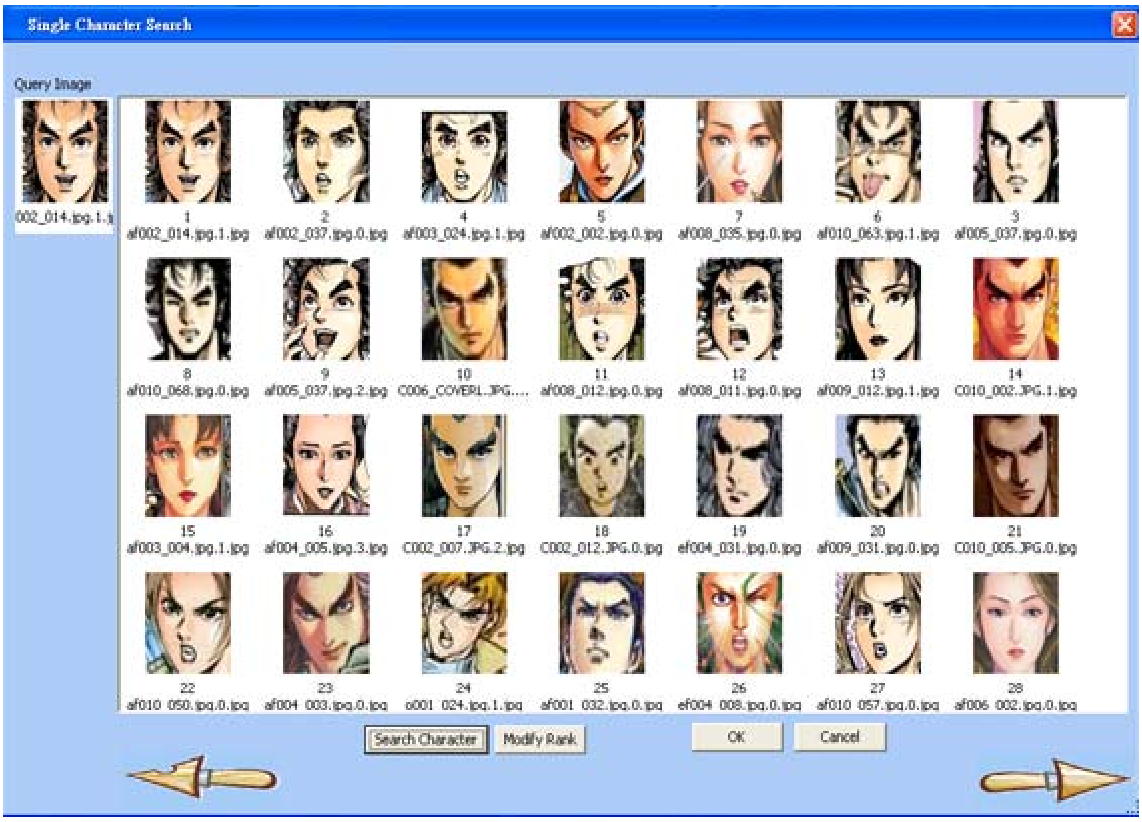


Figure 4.4.7b *The query result of a character who doesn't have any face images saved in the Bank*

4.4.8 Multiple Characters Searcher

If users want to search for a particular scene of different characters they can select the multiple characters search in 4.4.3. *Figure 4.4.8* is an example of the search result. Notice that none of the image faces in the query exists in the query result. But by using the multiple searcher we then would be able to obtain the image page where two of the query characters coexist.

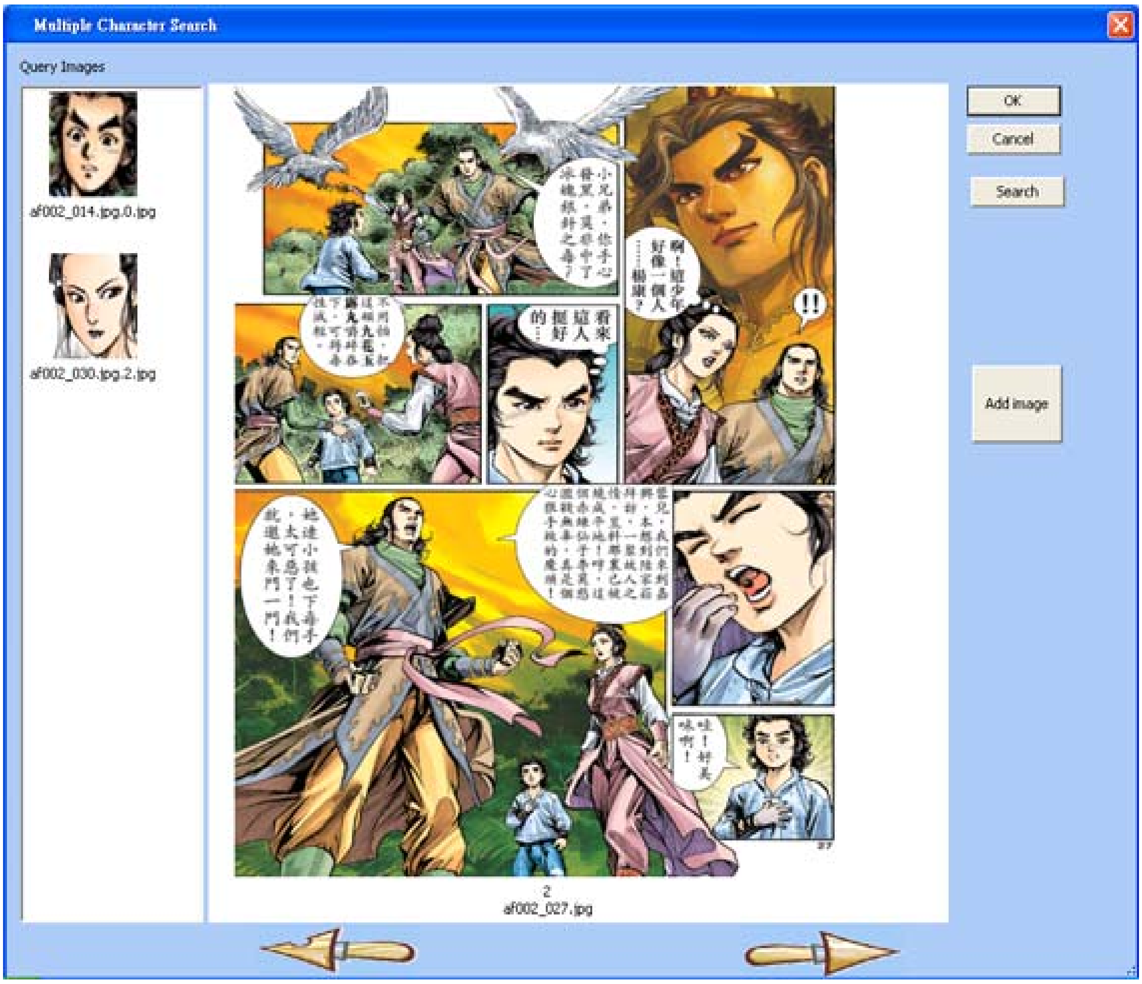


Figure 4.4.8 *Multiple Characters Searcher*

4.4.9 Help site

If users are confused of the progress of MAIRE, clicking the “help” button will take them to the help website for more information.

Figure 4.4.9 *Online Help Site*

4.4.10 Specifying EBGM Landmark Locations

Before performing EBGM recognition, the original design of EBGM requires to enter 25 landmarks; however, as entering all the landmarks for the entire set of training model might be unfeasible for users as this step is more tiring than brute-force searching their craved comic pages; moreover as the boosted EBGM performance is better than that with all the locations of the landmarks specified, this function is not furnished in the real release of MAIRE, but simply kept for research use.

So in the current application release the face model of 25 landmarks will be predefined automatically by the known information of the eye coordinates, the others could possibly be roughly estimated.

Figure 4.4.10 *Specifying EBGM Landmark Location after clicking on the right eye*

4.5 Design View of MAIRE to Cope with the Inaccuracy of Algorithms

Since the performance of detection and recognition algorithms will not work perfectly, it worths to have a little discussion on how the design of MAIRE can avail against the performance.

<u>Ground Truth Tool</u>

MAIRE is embedded with the face and eye tools for users to annotate the face details. So one may suspect if the detection part is that necessary to the application as the detection rate is not as accurate as it can be. In fact, even if the detection rate is not perfect, in reality it does save user's effort on manually annotating the faces on the comic pages. Actually, it is quite exhausting to annotate all the faces from scratch; to annotate faces on 1000 comic pages manually, it costs more than 6 hours; while it only costs users 2 hours to amend the results and obtain all the faces with the aid of face detection. Hence the ground truth tool is only served as a way to improve the results of detection, but not the pure solution to annotate all the faces.

Ranking

Recognition is a type of classification, in which the latter is well-known for classifying a test object to yes or no upon training. In presenting the results of the query, MAIRE could also be implemented as merely returning the comic faces which the distances of them from the query images are within a certain threshold. However, it is difficult to determine the threshold of distances. It ranges from the set of data set, the algorithms and the distance measures applied by the algorithms. Although the algorithms and the distance measures could be set by MAIRE, the distribution of data set, specified by the user, is unknown. So the threshold of distance is unpredictable.

Even if we have obtained a nice threshold that can classify the face image results into “same character as the query image” and “other characters”, and MAIRE displays all the face images that are within the distance threshold, due to the nature of classification problem, the result of the query is not always perfect. In turn this will be more difficult for users to find a desired comic face from the pool of wrong results.

Thus, instead of solely classifying a testing image as the same type of the query character, the results are displayed by rank. Ranking is also a popular way of presenting results from recognition by which it ranks all the testing images from the smallest distance to the larger distance. The testing images more likely to be as the same class of the query image will be of a higher rank. So by ranking, the threshold problem is solved, also, even if the recognition result is not that good, the user would still be able to retrieve his desired face image in a lower rank.

Modify rank list

Yet the recognition result is not ideal, the user can modify the list by using the "modify rank" dialog to save the characters of the same class and modify the rank of the search result. Upon modification, if the user uses MAIRE to search for that distinct character again, those face images which have been saved will appear as top ranks, consequently, the recognition result produced by MAIRE will become more and more accurate as users use it constantly.

Multiple characters search

It seems to be quite a difficult job for users to remember the exact comic face of a particular scene, which he wants to find, from hundreds and thousands of comic faces ranking results. Even if the single character search returns all exact face images of the searched character in top ranks cannot help the user to determine which face image is drawn from the scene he wants to find. That means simply the single character search is not powerful enough to achieve the objectives of MAIRE. So multiple characters search is implemented. By searching a list of comic characters, MAIRE will then be able to find the image pages that contains those characters, in turn narrowing down the scope of possible searched comic pages. As users most likely will remember who else are related to that scene, by entering all the different characters that are related to the scene, not only that it is easier for users to find what it wants, but also the performance of MAIRE on recognition is enhanced as the characters who has misclassified will not appear in the result. Say for example, if the user wants to search for 2 characters from different stories, MAIRE will return 0 results. Thus by specifying more characters related to the desired comic pages MAIRE in turn has more information on the scene the user wants to search, in addition to that the performance can be enhanced from simply single character search.

Chapter 5 -- Experimental Results and Discussion

In this chapter the experiments of the 7 algorithms described in chapter 3 will be conducted; discussion will be followed accordingly. From the results done by these experiments Adaboost and EBGM are the recommended algorithms to detect and recognize comic character faces.

5.1 Experiments on Face Detection

As mentioned in the Literature Review, detection methodologies can be classified into image and feature based approach. Thus some methods from both categories will be tested in a set of comic pages to investigate the performance.

To compare the results from different algorithms, initially the ground truth of the faces from the data set is obtained, by examining if the "detected faces" lie roughly on the coordinates provided by the ground truth, the 2 vital elements in evaluating the accuracy of the result, true positive (actual faces) and false positive (false detected faces), can be determined.

5.1.1 Experimental Setup

5.1.1.1 Data Set

104 e-comic pages are extracted from 2 sets of comics, CondorHeroes (神鵰俠侶) and BiohazrdProjectx (生化危機 Project X), containing 413 faces overall.

5.1.1.2 Assumption

- All the "faces", consisting of the major and minor characters, are taken into an account of a “face”. So all kinds of blobs are assumed to be faces, no matter if they are ambitious, blurred or occluded.

5.1.2 Low Level Analysis – Skin Color Segmentation

5.1.2.1 Determining the Color Space

The common color spaces for testing are RGB, HSV, YCbCr and LAB. The results are listed in *Table 5.1.2.1*.

5.1.2.2 Filtering of False Detected Faces in HSV

As the false positive rate is too high to accept, filtering is to be done as mentioned in section 3.1.2; and the corresponding final result is shown in *Table 5.1.2.2*, where the triangle denotes the result percentage of the real application.

5.1.2.3 The Result of Skin Color Segmentation

From the receiver operating characteristic (ROC) curve in *Figure 5.1.2.3*, it can be seen that HSV would provide the best performance. Thus HSV are selected to be included in the application of this project.

For comic data set, the advantages for using skin color detection for faces are:

- Faces of varies poses could be detected if the face color lies on the specified skin color region.
- The majority of comic faces are in the same color region, it is not needed to deal with various ethnicities of faces like skin color detection for real-world images.

However, the problems still remain as:

- A good color space and threshold have to be determined

- Occasionally there are comic faces of non-skin color
- The results include some regions of the skin color even after filtering has been completed (e.g. hands, pink backgrounds)

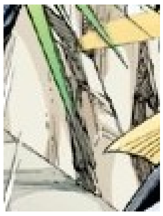
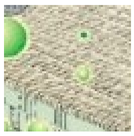
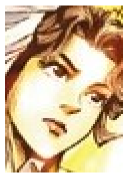
Color Space	Accuracy	False Detect	Miss
RGB	86.4% 	69.12% 	13.6% 
HSV	88.3% 	60.20% 	11.6% 
YCbCr	84.0% 	89.2% 	16.0% 
LAB	85.0% 	86.4% 	15.0% 

Table 5.1.2.1. *The face segmentation result for different color spaces*

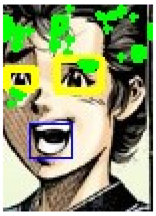
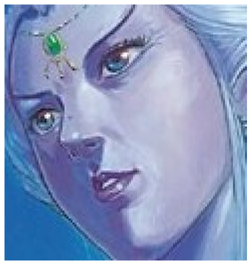
	Accuracy	False Detect	Miss
Final result	70% 	70% 	30% 

Table 5.1.2.2. *The result after filtering*

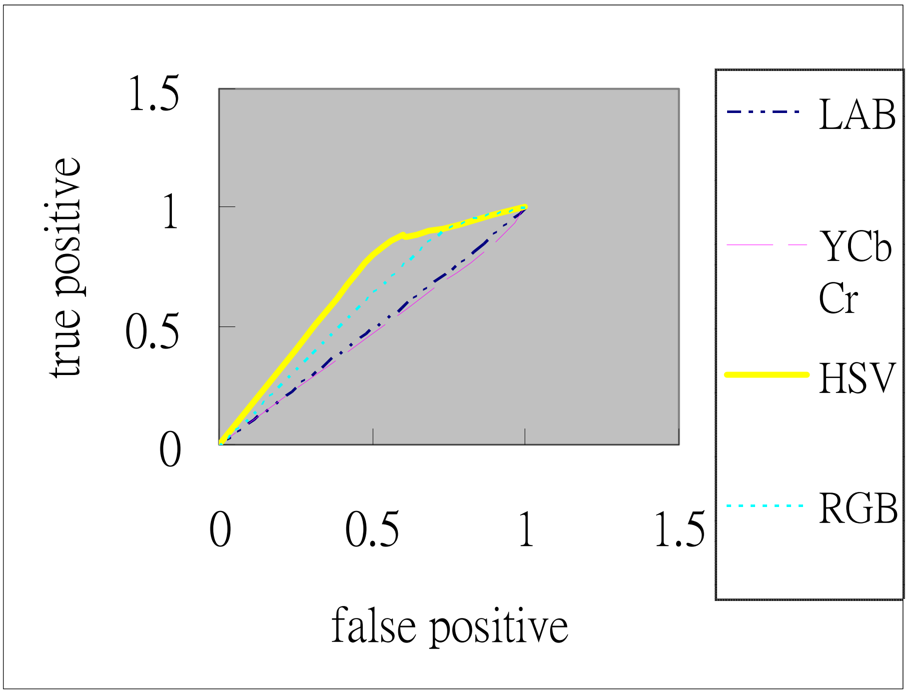


Figure 5.1.2.3. *ROC Curve for the 4 color spaces*

5.1.3. Image based Approach

The result of Neural Networks and Adaboost, as described in section 3.1.3 and 3.1.4, are revealed in *Table 5.1.3* and the corresponding ROC curve is shown on *Figure 5.1.3.*

	Accuracy	False Detect	Miss	Running Time
Neural Networks	43.3% 	64.0% 	56.7% 	~9 hours
Adaboost	45.5% 	35.2% 	54.5% 	~30 minutes

***Table 5.1.3** The detection results of Neural Networks and Adaboost*

The Adaboost performance is not as good as the result from other researches, although it is a state of the art methodology that had been applied on loads of face detection scenario. The reason behind this is that during in the training process, a large number of data set has to be obtained, for both face and non-face. And if each pose of the face had to be trained by say 1000 pose images, and in the comic sets usually exhibits more than 4 poses, one pose for a cascade, each strong classifier ordinary takes 3 weeks of full computation time to train, so at least 12 weeks of merely training is required, not including the optimal way for training a better classifier mentioned in section 3.1.3.1. Thus here a simplified version is used by applying one cascade, so undoubtedly, the performance is not as good as the research already done.

The Neural Network had a low performance could be because of “the system is only able to detect upright, frontal faces” [17], which are not the majority of the faces occurred in comics images.

5.1.4 HSV Segmentation VS Adaboost

As the Skin Color Segmentation by HSV color space and Adaboost are the best ones we have experimented, we would like to see which of them fits better in the CBIR system.

Comparing the yellow triangle and the pink square applied in MAIRE in *Figure 5.1.3*, Adaboost can detect less true positive than skin color segmentation, but the false positive rate is not that high. So as it is easier to complete a delete operation in the user interface design of MAIRE rather than adding in or modify the coordinates of the face features, some users could prefer the skin color detection; however generally speaking Adaboost is better at localization[2], and as some of the "faces" that are ambiguous are not necessarily to be detected, thus Adaboost is promoted to be used. But it really depends on need of users and the set of the input comic images.

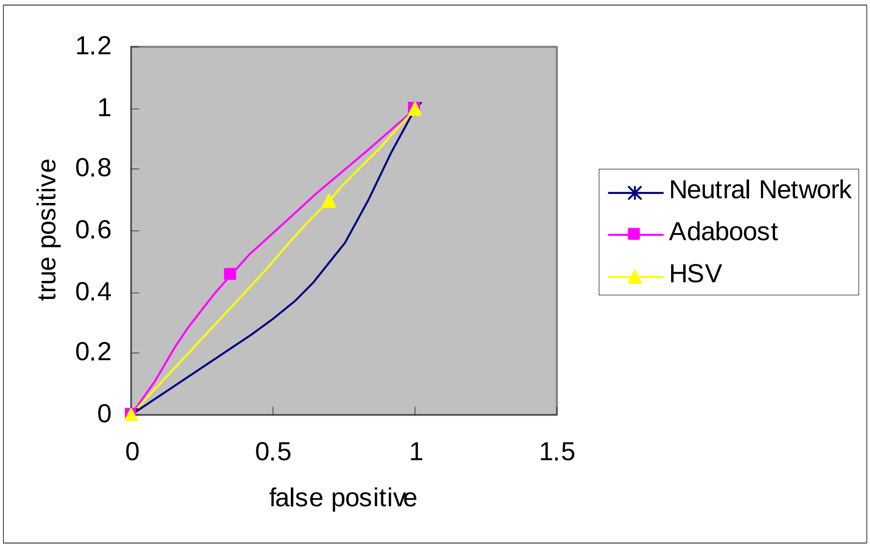


***Figure 5.1.3** ROC curve for the experimented methodologies*

[2] HSV detection will include unnecessary features in annotation like neck

5.2 Experiments on Face Recognition

Recognition experiments are conducted on the set of comic characters' faces by applying different algorithms. With the obtained data set, half of the images will be used for testing and the other half will be for training, the images are selected by random sampling. The results are recorded and the experiments are performed again with another set of training and testing set. The overall averaged result from these experiments are presented in this report.

5.2.1 Experimental Setup

5.2.1.1 Data Set

The Data Set is drawn from the outcome of Adaboost detection with manual amendment and annotation. 1013 images are obtained[3] from 5 stories including CondorHeroes (神鵰俠侶), DragonTiger5Generation1(龍虎 5 世), DragonTiger5Generation2 (龍虎 5 世V), Firemen (滅火群龍) and BiohazardProjectx (生化危機 Project X). There are 79 classes with more than 1 face image and 187 single-class-image, which can be classified as outliners.

To have more premier knowledge on the data set, both *Figure 5.2.1.1a* and *Figure 5.2.1.1b* show some distribution of the data from the comics, and the class and class-comic distribution. The lexicographical order in the latter figure corresponds to the sequence of comic sets shown in the former graph.

[3] Some of the imageries are appendixed on Appendix B

5.2.1.2 Assumption

- Every comic set has its own set of characters which will not overlap with other comic sets.

  The comic set DragonTiger5Generation1 and DragonTiger5Generations2 seems to be prequel and sequel. But here they are treated as different set of comics[4], so some of the characters in either class may look similar as the other in a way that the sequel faces exhibit more grown up features. So as they are identified as different comic sets, these types of characters are classified as 2 classes, one class in the sequel and one from prequel.
- Within the same comic story, the child faces and the grown up faces are classified as the same character if these two types of face images are representing the same comic character.

  For some of the comic stories such as CondorHeroes, the plot focuses a large portion in describing the life of a character, so although the character may look not exactly the same when the comic chapter proceeds, they are to be classified into the same character.

[4] This is because it is always not likely for the plot of the sequel to be related to the prequel, that is simply some of the background and the sequel is more likely to be a set of other story.

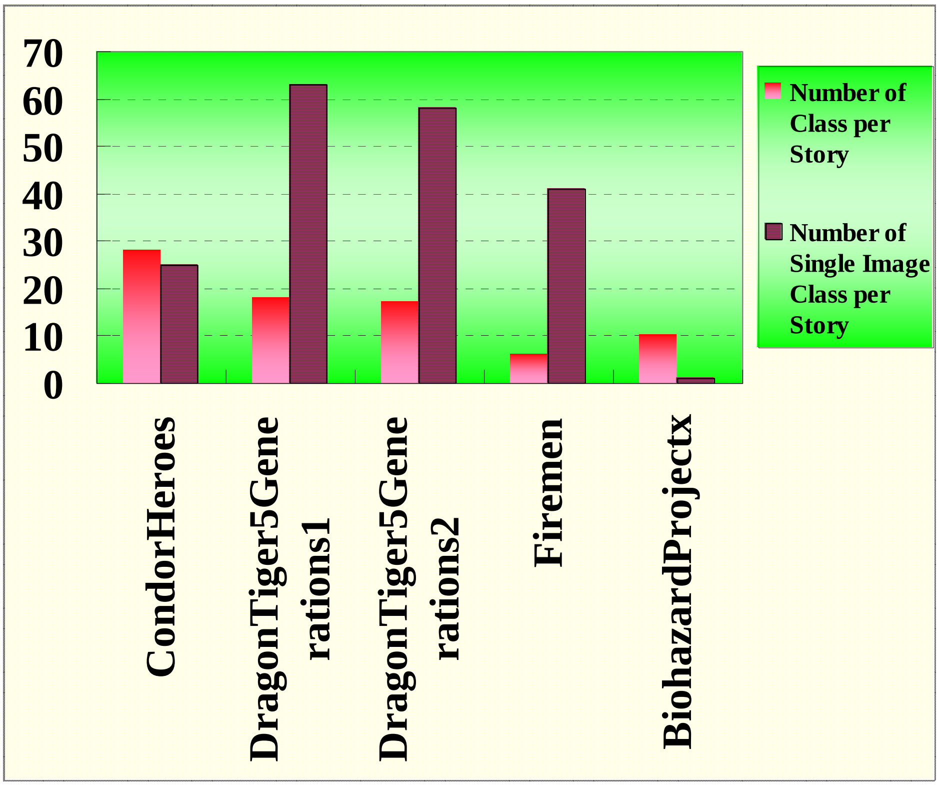


Figure 5.2.1.1a *class characters distribution of the comic sets*

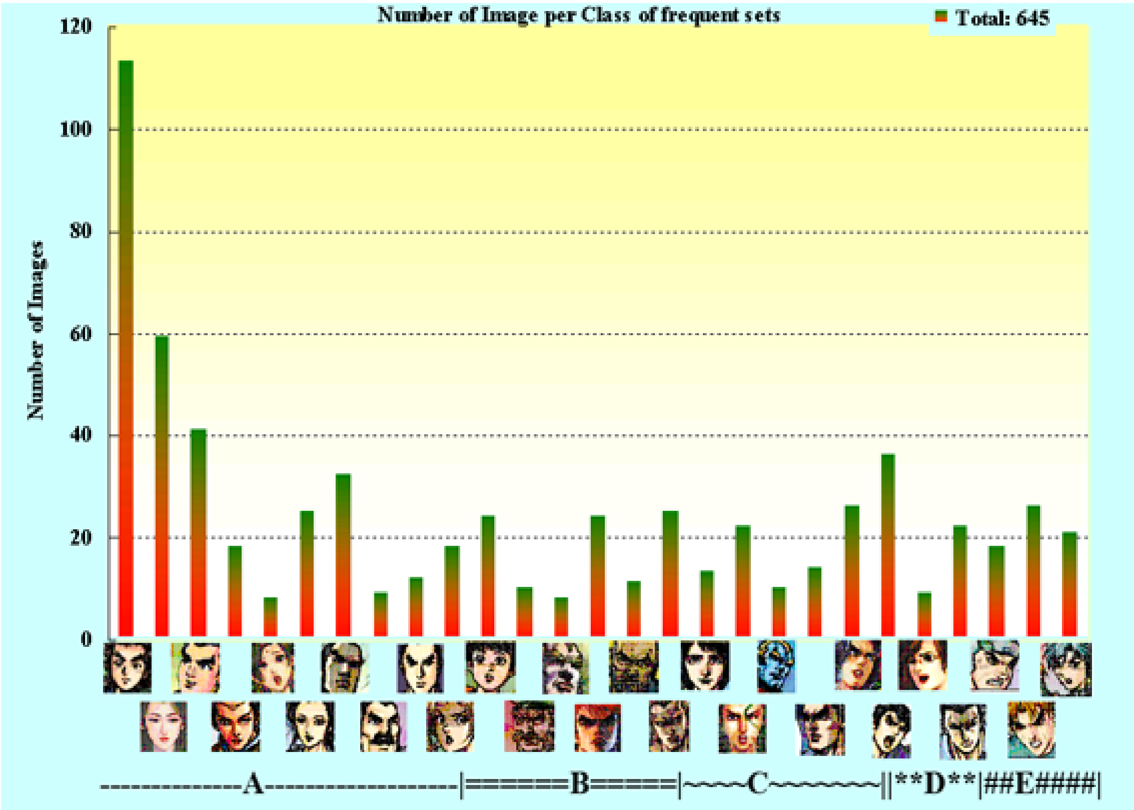


Figure 5.2.1.1b *Class and class-comic distribution of the data set*

(only the most frequent classes are shown)

5.2.2 PCA and LDA Distance Measure

Experiments are conducted on applying different distance measures on PCA and LDA to obtain the best one that can work on the comic data set.

From *Figure 5.2.2a* and *Figure 5.2.2b*, PCA YamborAngle and LDA Covariance are the most applicable distance measures in PCA and LDA on recognizing the comic image faces.

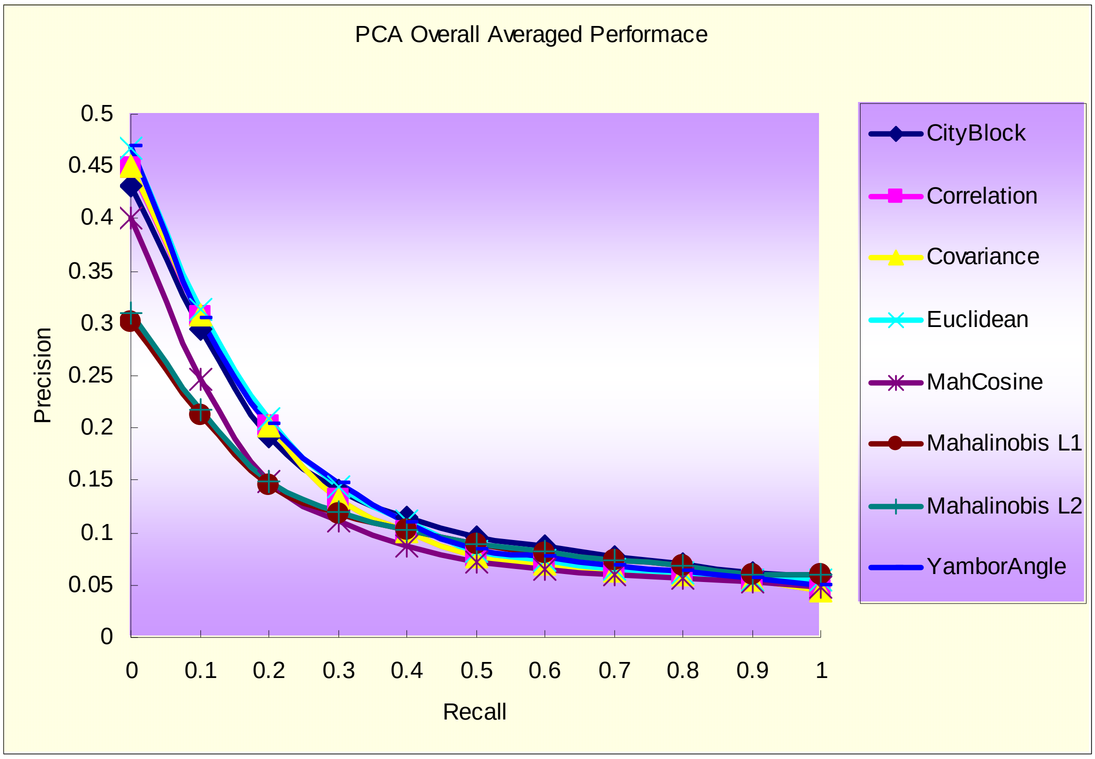


Figure 5.2.2a *Overall PCA Performance on different distance measures*

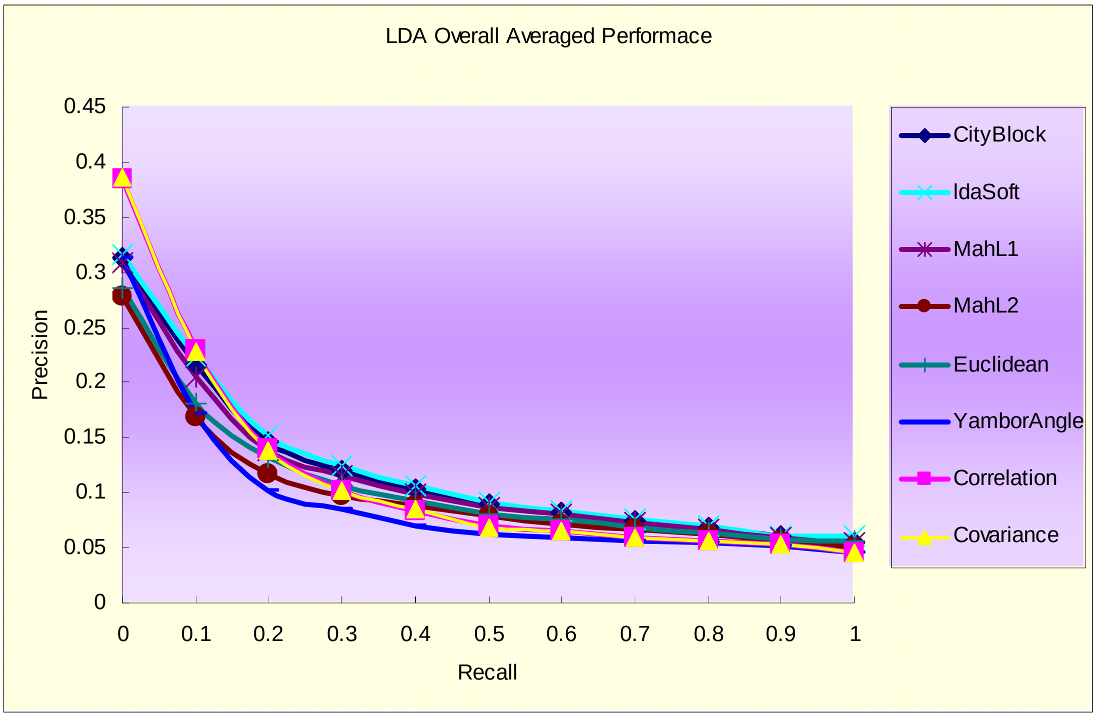


Figure 5.2.2b *Overall LDA Performance on different distance measures*

5.2.3 Overall Performance

The figure below (*Figure 5.2.3a*) shows the overall performance of the 4 different recognition algorithms.

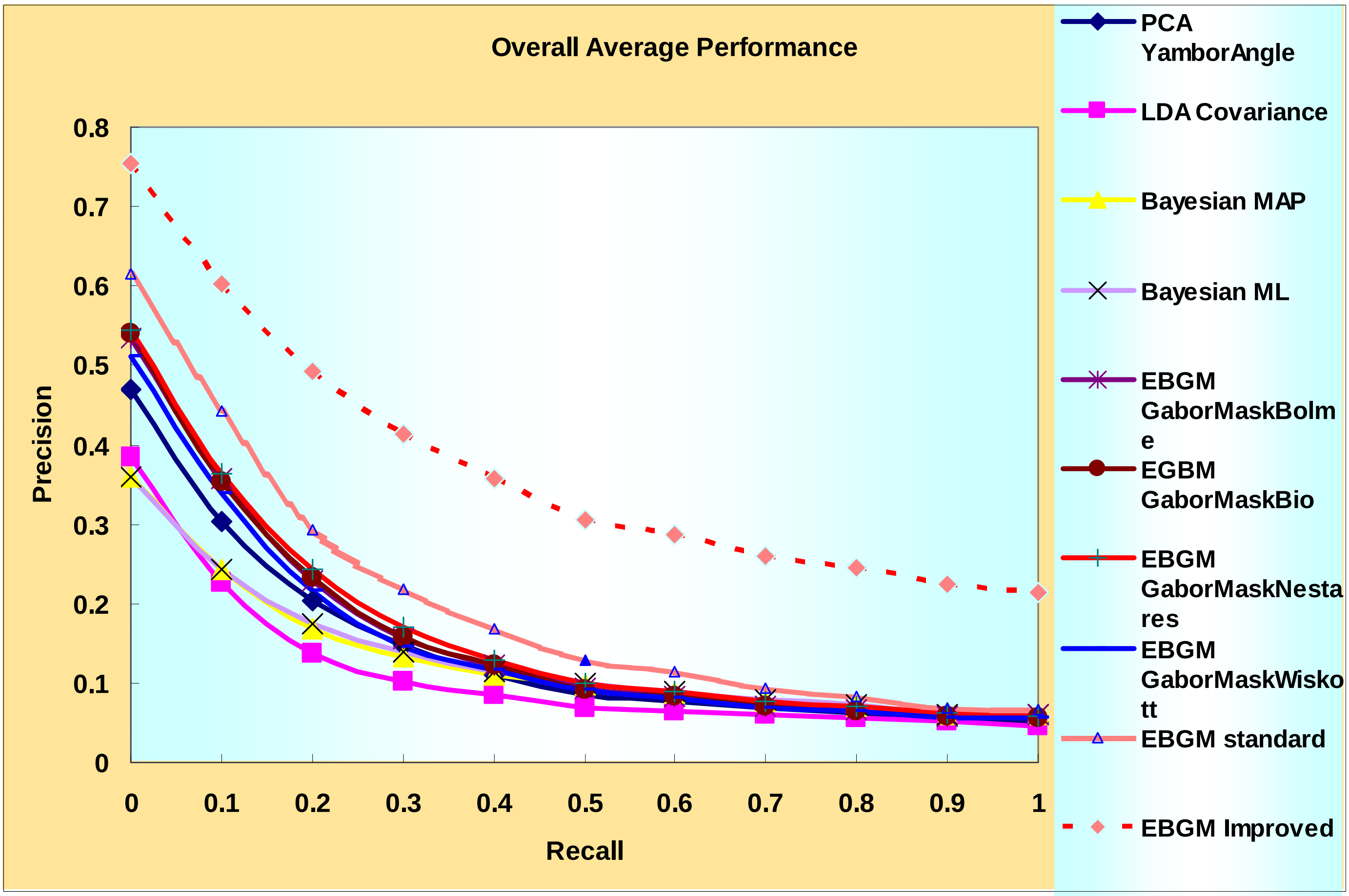


Figure 5.2.3a *Overall Averaged Performance by the 4 recognition algorithms applied*

PCA YamborAngle and LDA Covariance will give the best results in the PCA category and LDA category by the above section, so they are selected to be compared with other algorithms.

It is shown that Bayesian Intrapersonal/Extrapersonal Classifier, for both maximum a posteriori (MAP) and maximum likelihood (ML), have poor performances, where LDA, as well, doesn't give us a better outcome. The best algorithm on the comic set is shown to be EBGM, with specifically outstanding retrieval results by the Nestares Wavelets.

The EBGM standard curve is drawn by manually locating all the 25 landmarks from the whole set of images instead of using the normalized ones, by using the Nestares Wavelets. Undoubtedly, it gives a better result than EBGM GaborMaskNestares (which simply locates all the landmarks by estimation); however, it takes around 3 minutes to locate all the points on a face image, which makes it half a day on solely locating the entire set of the training data if the training size is huge, this is quite impractical for real-life application as it is an exhausting job to locate all features.

Since the whole set of images are cropped from different sets of comics, the overall result can be enhanced by first retrieving the images that are cropped from the same story of the query image, which is the EBGM improved curve in *Figure 5.2.3a* by applying the Nestares Wavelet. It has shown that this can improve the performance of EBGM Nestares significantly, in which the result is even much better than exhaustive finding all the 25 landmarks manually. Thus, by applying the improved EBGM Nestares to MAIRE, not only the tough work could be saved, but also the performance is ameliorated by 38.79%.

By comparing the improved curve and EBGM Nestares curve, more information regarding to the data set can be obtained. As the Nestares curve only drawn on the face graphs to compute the ranking but the improved curve is facilitated by where the test face graphs are from, one can say that a number of characters in a comic will have similar features from another comic, of which the sequel and prequel can prove this.

It is quite logical to see that EBGM towers over the other algorithms. The features represented by EBGM are the Gabor wavelets, which are well known for multi-resolution analysis. By the extra properties of Gabor wavelets (such as being able to accommodating frequency and position simultaneously), it suits perfectly into the face recognition problem which varies at different scales or resolution. Besides Gabor wavelets, the characteristics of face graphs also contribute to the effectiveness of EBGM. Not only until the face graphs with fiducial points of the face images are obtained, it is difficult to determine where to locate the Gabor wavelets for feature extraction. As the landmarks are located onto facial features such as the eyes and mouths, which are more distinctive features of faces, the Gabor wavelets then can select the features located on the respective landmarks for comparison, in turn determining which images are more likely to be classified as the same type of the query image.

Obviously, PCA does not perform as good as EBGM, this result is expected as PCA does not make use of the intrinsic features of a face to recognize; however it at least outperforms LDA.

LDA operates by finding a projection which maximizes the ratio of distances between classes and distances within classes, so it sounds like LDA will have a better performance than PCA, which simply finds a subspace with vectors corresponding to the maximum variance directions in the original space. However, as the comics data set is not that large and it consists of varies number of class sizes and outliners, the performance of LDA is not as good as PCA as the latter is less sensitive to different training datasets[25]. *Figure 5.2.3b* is an illustration of the scenario when PCA outperforms LDA.

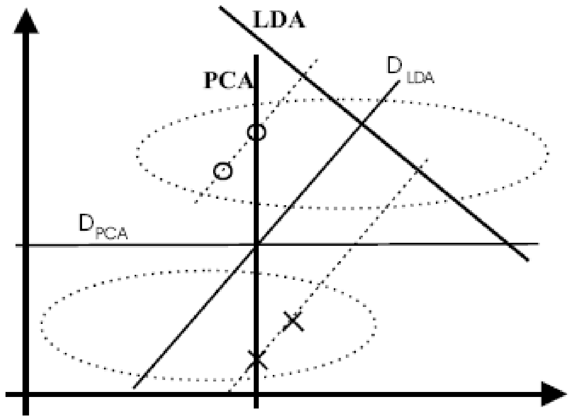


Figure 5.2.3b *Situation when PCA outperforms LDA [25]*

The less perfect result of Bayesian classifiers could be due to the undesirable distribution of Gaussian generated by the parameterized by PCA.

Opting for finding out what factors affect the result of recognition of EBGM, the following analyses are done based on the EBGM standard curve in *Figure 5.2.3a*.

5.2.4 Cartoonist and Story plots

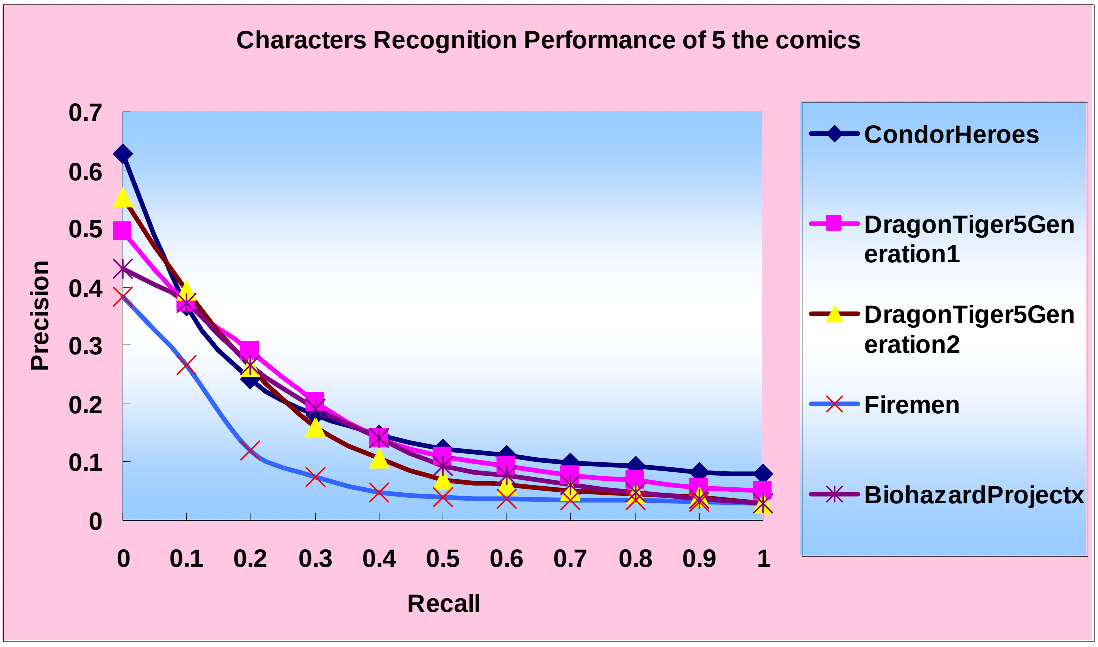


Figure 5.2.4 *Characters Recognitions Performance of the 5 comics*

Next, the performances of different set of comics are plotted as above (*Figure 5.2.4*). By observation, it is believed that the recognition performance could have been affected by the cartoonists. CondorHeroes and DragonTiger5Generations1&2 are drawn by Mr. Wong (黃玉朗); where Mr. Chan (陳偉文) and Mr. Lam (林謂康) are the cartoonist of BioharzardProjectx and Firemen correspondingly. All comics by Mr. Wong have a better recognition result, this could be related to the design of the comic characters, most of the characters in the first 3 comics carry more distinguishable features.

In fact the characters in Firemen also carry distinguishable features, but owning to the story plot is about firemen, who wears yellow helmets in most of the scenes and half of the head features are covered up, EBGM might not be able to accomplish as good as the non-firemen.

For another type of comic faces which have less distinguishable features, as in BiohazardProjectx, it is expected that the recognition rate is not as good as those exhibits more facial feature, for example, the characters in this set of comic does not have any nostrils, where most of the characters drawn by Mr. Wong have these features.

5.2.5 Occluded

Figure 5.2.5a shows the recognition rate of some of the occluded faces in the data set.

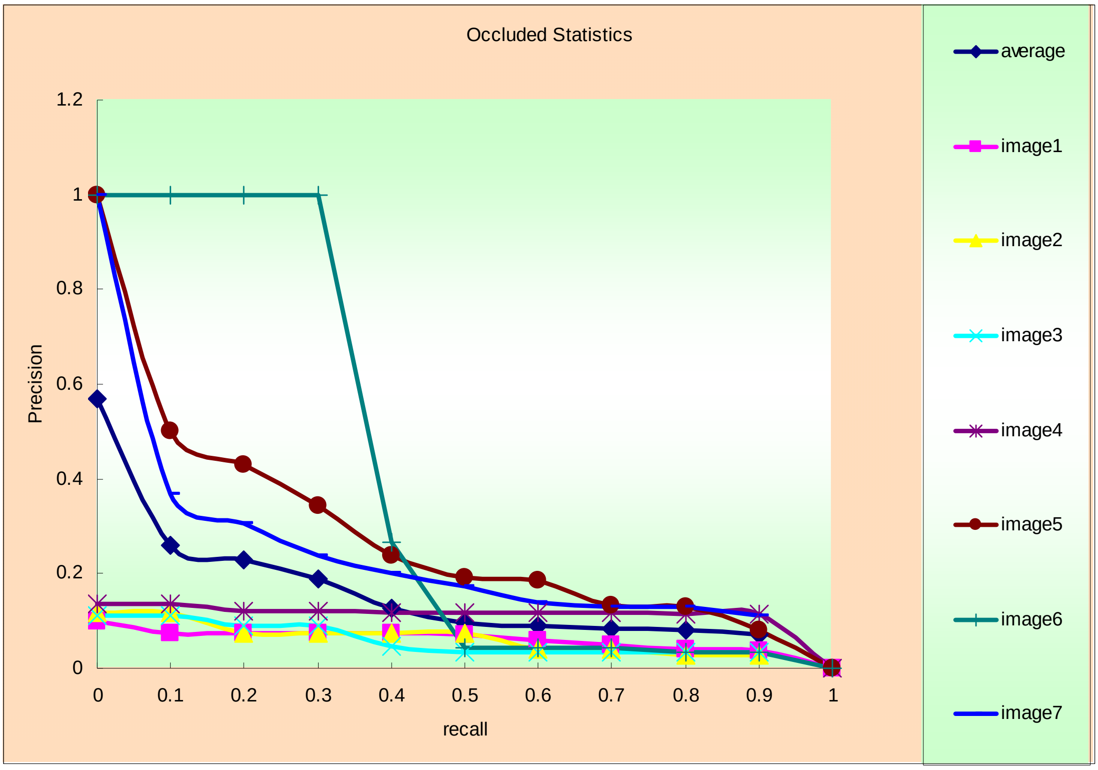


Figure 5.2.5a *Occluded Statistics*

Image1	Image2	Image3	Image4	Image5	Image6	Image7
						

Table 5.2.5b *Mapping of faces to Figure 5.2.5a*

It can be seen that not all occluded faces will lead to poor recognition result. The result of recognition of occluded faces will only be affected by how occluded the query images are, or what type of features of face image is missing.

Occluded face in Image5, Image6 and Image7 only have minor missing features in the chin, left head and the top head region respectively, but still they are able to get good recognition result.

According to Image4, the right eye is cut off and the right head is also a bit occluded.

Compared with Image6, of which only the left head is cut off, Image4 has far low performance on recognition. Thus it is believed that the eyes are very deterministic feature in EBGM in providing a good classification.

Image1, Image2 and Image3 also give undesirable classification result. This is because most of the features of the faces are missing: in Image1 we can only view the left eye, left eyebrow and left head; in Image2 although eyes, nose and mouth are present, the chin and every outline of the head is missing. So when too many information of the face image are missing, it is difficult for the algorithm to do the matching job.

5.2.6 EBGM Class Characters View

The following graph shows the performances of the class characters by the Nestares wavelets.

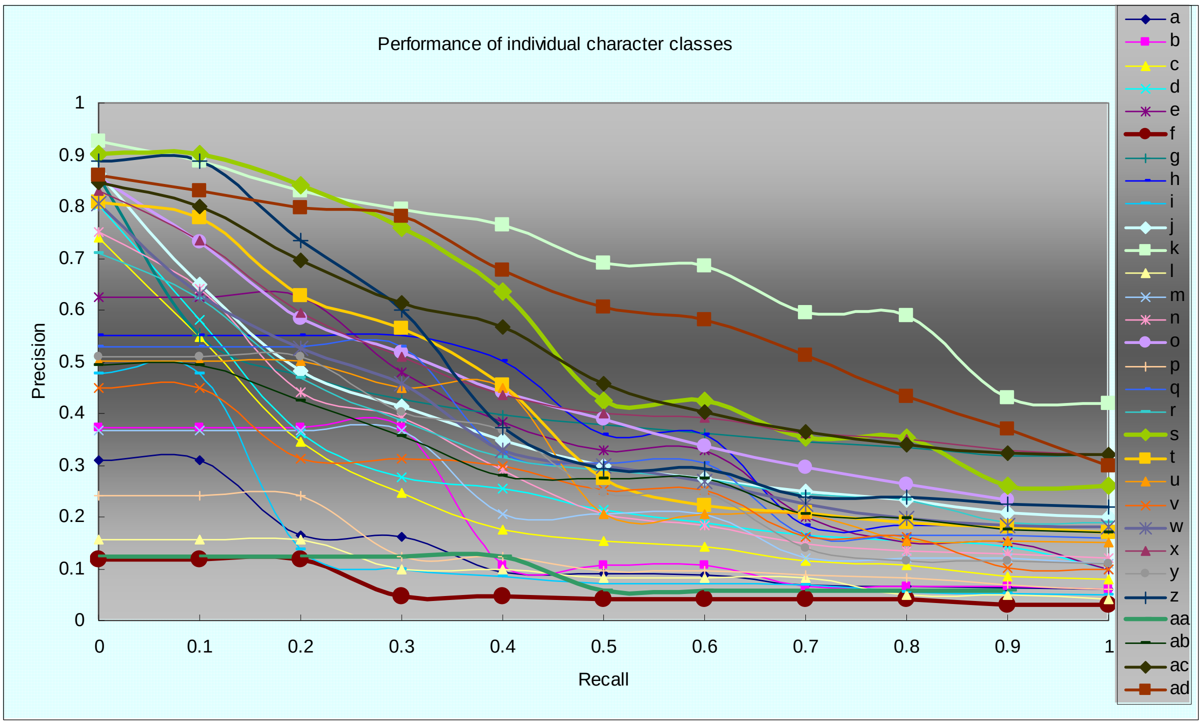


***Figure 5.2.6a** Performances of individual class character*

Legend mapping:

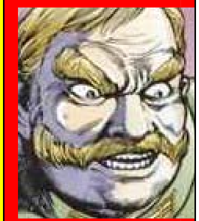
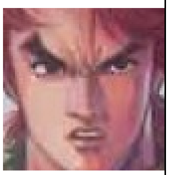
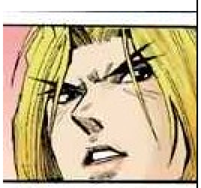
a 	b 	c 	d 	e 	f 
g 	h 	i 	**j** 	**k** 	l 
m 	n 	o 	p 	q 	r 
s 	**t** 	u 	v 	w 	x 
y 	**z** 	aa 	ab 	ac 	**ad** 

Table 5.2.6b *Legend mapping for Figure 5.2.6a; the characters highlighted in red and underlined has performed badly; where the ones bolded (in yellow) have a good recognition rate(>0.8)*

From this graph, it can be noticed that a majority of the well behaved classes are the female characters; and out of expectation, the characters that exhibit more intrinsic facial feature (aa, l and p) cannot be recognized by EBGM in top most ranks; it is believed that the more distinctive features a character exhibits, the easier it is for recognition; also notice that all of the 4 red highlighted characters are not the frequent set list in *Figure 5.2.1.1b*, so the poor performance might due to the lack of Gabor jets in the bunch graph to represent enough feature of the character.

Also, apparently, the performance of a class is not affected by the number of training data used in the training stage; s, k and ad do not have a lot of training data to train on (*Figure 5.2.1.1b*), but they could be recognized by EBGM rather accurately. So it is believed that once we have obtained a good enough model graph to represent the character, and provided that the query image could fit into the model nicely, the recognition result will be quite ideal.

5.2.7 Images with Low Performances on EBGM

Apart from the understanding the behavior of EBGM by class views, there might be some shared features in the classification of the query images which lead to poor recognition rate. Below shows the images of which the recognition by EBGM Nestares are extraordinary coarse (with lower than 0.05 precision).

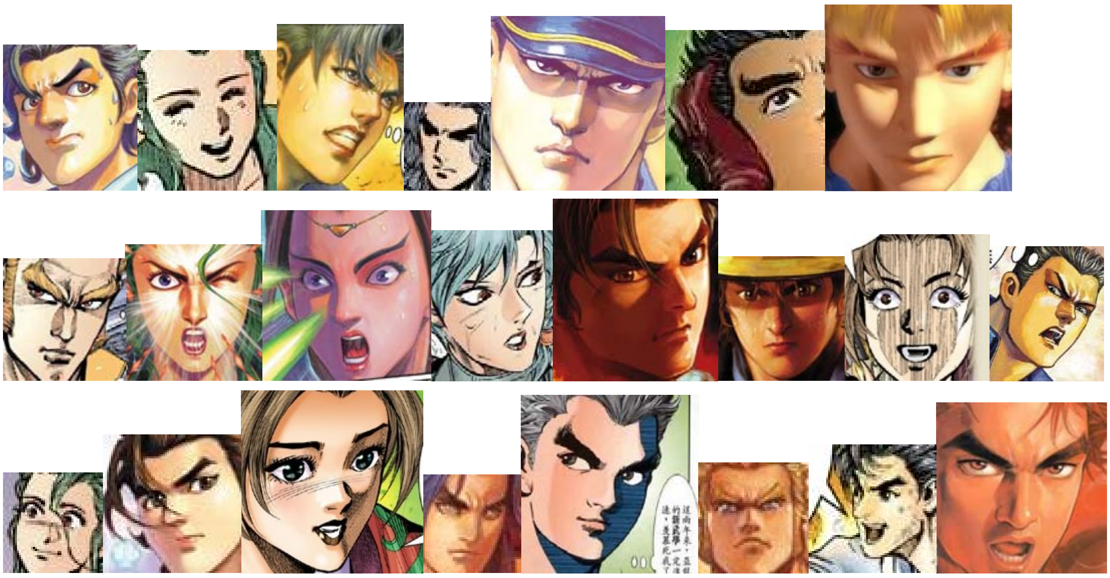


Figure 5.2.7 *Query Image with low recognition rate by EBGM Nestares*

It can be easily spotted out that a majority of these images, surprisingly, are in form of full-colored spread imageries, which most of them are cropped from the comic cover or poster pages. Thus, these images are believed to be the attribute for lowering the performance of recognition.

To deal with this kind of situation, this kind of full-colored imageries should have been trained by EBGM in the localization stage. But since these kind of nicely colored comic pages are not frequent (3-5 pages from a volume of 30 pages), along with the characters on these pages mostly focus on the main characters, it is difficult to obtain these kind of training data for other characters. It is shown that if these kinds of imageries are trained in the training stage, EBGM would be able to recognize those faces of the character in full-colored spread.

Chapter 6 -- Conclusion and Future Work

6.1 Critical Review

To cater for the need of comic readers to dodge from exhaustive searching for a particular scene in a set of comics, MAIRE, a CBIR system for viewing and searching comic characters is introduced in this project.

CBIR operates on image, which is difficult to be comprehended by computers in form of a matrix of pixels. To them the matrixes are simply numbers corresponding to which color should be displayed on the respective coordinate on the screen. Thus it is not an easy task for machines to distinguish faces and characters in an image and there is no guarantee of 100% accuracy although various of algorithms are furnished to complete the task.

Comparing different techniques to apply on MAIRE by experimenting the performances on the comic data set, the most workable techniques for detection and recognition are found to be Adaboost and EBGM correspondingly. Despite of their imperfectness, the results are still reasonable to be employed on the application for deployment. To cope with the inaccuracy from machine vision, modification of the results can be facilitated by MAIRE.

The results of this study may provide an additional channel for comic readers to ease their searches of character images by narrowing down their searching area to a particular scene. Hopefully the arise of MAIRE not only can fulfill users' need, but can also advertise the popularity of e-comics over the traditional paper volumes so as to save more trees.

6.2 Further Development

Hopefully this research is able to open up new vistas to searching and identifying characters in “video comics” like cartoons.

References

Cited References

[1]. Hjelmas, E and Boon K.L. (2001). Face Detection: A Survey. Computer Vision and Image Understanding 83, 236–274.

[2]. Sakai et al. (1972). Computer analysis and classification of photographs of human faces. In Proc. First USA—Japan Computer Conference, 55-62.

[3]. Silva, D. et al. (1995). Detection and Tracking of Facial Features. Proc. of SPIE Visual Communications and Image Processing: Taiwan.

[4]. Sung and Poggio. (1998). Example-Based Learning for View-Based Human Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence archive 20 (1), 39-51.

[5]. Rowley, H., S. Baluja and T. Kanade.(1998). Neural Network-based Face Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1), 23-38. [Online]. Available: http://www.ri.cmu.edu/pub_files/pub1/rowley_henry_1998_1/rowley_henry_1998_1.pdf

[6]. Yuille, A. L., Cohen, D. S., and Hallinan, P. W. (1992.) Feature extraction from faces using deformable templates. Int. J. Computer Vision 8, 99–112.

[7]. Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. IEEE Trans. Patt. Anal. Mach. Intell. 23, 681–685.

[8]. Zhao, W., Chellappa, R., Phillips, P. J. and Rosenfeld, A. (2003). Face Recognition: A Literature Review. ACM Computing Surveys 35 (4), 399-458.

[9]. Lu Xiaoguang. Image Analysis for Face Recognition. [Online]. Available: http://www.face-rec.org/interesting-papers/General/ImAna4FacRcg_lu.pdf

[10]. Wiskott, et al. (1997). Face recognition by elastic bunch graph matching. IEEE Trans. Patt. Anal. Mach. Intell.19, 775–779. http://www.face-rec.org/algorithms/EBGM/WisFelKrue99-FaceRecognition-JainBo

ok.pdf

[11]. Blanz et al. (1999) A Morphable Model For The Synthesis Of 3D Faces. Computer Graphics Proceedings, [Online]. Avaliable: http://www.mpi-inf.mpg.de/~blanz/html/data/morphmod1.pdf

[12]. Kapur, J P. (1997). Face Detection in Color Images. EE499 Capstone Design Project Spring. University of Washington Department of Electrical Engineering, [Online]. Avaliable: http://www.geocities.com/jaykapur/face.html

[13]. Singh, S. Kr, ChauHan, D.S., Vatsa, M, and Singh, R. (2003). A Robust Skin Color Based Face Detection Algorithm. Tankang Journal of Science and Engineering , 6(4), 227-234. [Online]. Avaliable: http://www2.tku.edu.tw/~tkjse/6-4/6-4-6.pdf

[14]. Viola and Jones. (2001). Robust Real-time Object Detection. International Journal of Computer Vision. [Online]. Avaliable: http://lear.inrialpes.fr/people/triggs/student/vj/viola-ijcv04.pdf

[15]. Lienhart, R., Kuranov and Pisarevsky, V. (2003). Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. *DAGM'03, 25th Pattern Recognition Symposium*, Madgeburg, Germany, 297-304.[Online]. Avaliable: http://www.lienhart.de/Publications/DAGM2003.pdf

[16]. Freund, Y. and Schapire, R.E. (1999). A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, 14(5), 771-780. [Online]. Available: http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf

[17]. Reuvers, M. (2005). Face Detection on the INCA. Philips Research Laboratories Eindhoven, Digital Design and Test Group, University of Amsterdam, [Online]. Available: http://www.science.uva.nl/research/ias/alumni/m.sc.theses/theses/MartijnReuvers.pdf

[18]. Guo, G-D. and Zhang, H. (2001). Boosting for Fast Face Recognition. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems (RATFG-RTS'01), in conjunction with ICCV'01, 96-100. [Online]. Available:

http://ieeexplore.ieee.org/iel5/7480/20325/00938916.pdf?isnumber=20325&arnumber=938916

[19]. Bolme, D.S., Beveridge, J.R., Teixeira, M. and Draper, B.A. (2003). The CSU Face Identification Evaluation System: Its Purpose, Features, and Structure. ICVS 2003: 304-313. [Online]. Available: http://www.cs.colostate.edu/evalfacerec/papers/icvsC0122.pdf

[20]. W.S. Yambor. (2000). Analysis of PCA-based and Fisher Discriminant-Based Image Recognition Algorithms, M.S. Thesis, Technical Report CS-00-103, Computer Science Department, Colorado State University, [Online]. Available: http://www.cs.colostate.edu/evalfacerec/papers/pcaldamixmod.pdf

[21]. Geof H.G., Beveridge, J. R., Draper, B.A. and Bolme, D.S. (2003) Using a Generalized Linear Mixed Model to Study the Configuration Space of a PCA+LDA Human Face Recognition Algorithm. [Online]. Available: http://www.cs.colostate.edu/evalfacerec/papers/pcaldamixmod.pdf

[22]. Moghaddam and Pentland. (1998) Beyond Eigenfaces: Probabilistic Matching for Face Recognition. Proc. of Int'l Conf. on Automatic Face and Gesture Recognition (FG'98).

[23]. Teixeira, M. (2003) The Bayesian Intrapersonal/Extrapersonal Classifier. 3rd International Conference on Computer Vision Systems - ICVS 2003, [Online]. Available: http://www.cs.colostate.edu/evalfacerec/papers/teixeiraThesis.pdf

[24].Bolme, D.S. (2003) Elastic Bunch Graph Matching. Colorado State University, [Online]. Available: http://www.cs.colostate.edu/evalfacerec/papers/EBGMThesis_Final.pdf

[25]. Martinez and Kak. (2001). PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (2), 228-233, [Online]. Available: http://cobweb.ecn.purdue.edu/~aleix/pami01.pdf

Online Background Web References

[26].

http://www.cs.utexas.edu/~grauman/courses/spring2007/395T/slides/Rapid_Object_Detection.pdf

[27]. http://www.robots.ox.ac.uk/~az/lectures/cv/adaboost_matas.pdf

[28]. http://franck.fleurey.free.fr/FaceDetection/

[29]. http://www.csd.uwo.ca/~olga/Courses//CS434a_541a//Lecture7.pdf

[30]. Eigenface Tutorial.
http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm

[31]. http://www.csd.uwo.ca/~olga/Courses//CS434a_541a//Lecture8.pdf

[32].
http://www.stat.ucla.edu/~sczhu/Courses/UCLA/Stat_231/Lect_note_2005/Lect4-6_Dim_reduction.pdf

[33].
http://www.stat.ucla.edu/~sczhu/Courses/UCLA/Stat_231/Lect_note_2005/Lect2_Bayes_decision_theory.pdf

[34].
http://www.stat.ucla.edu/~sczhu/Courses/UCLA/Stat_231/Lect_note_2005/lect12_13_Adaboosting.pdf

[35]. http://vismod.media.mit.edu/vismod/demos/facerec/feret.html

[36]. http://www.navyrain.net/facedetection.html

Implementation References

[37]. Open Source Computer Vision Library

[38]. Professional UI Solutions

[39]. The Code Project

[40]. CodeGuru.com

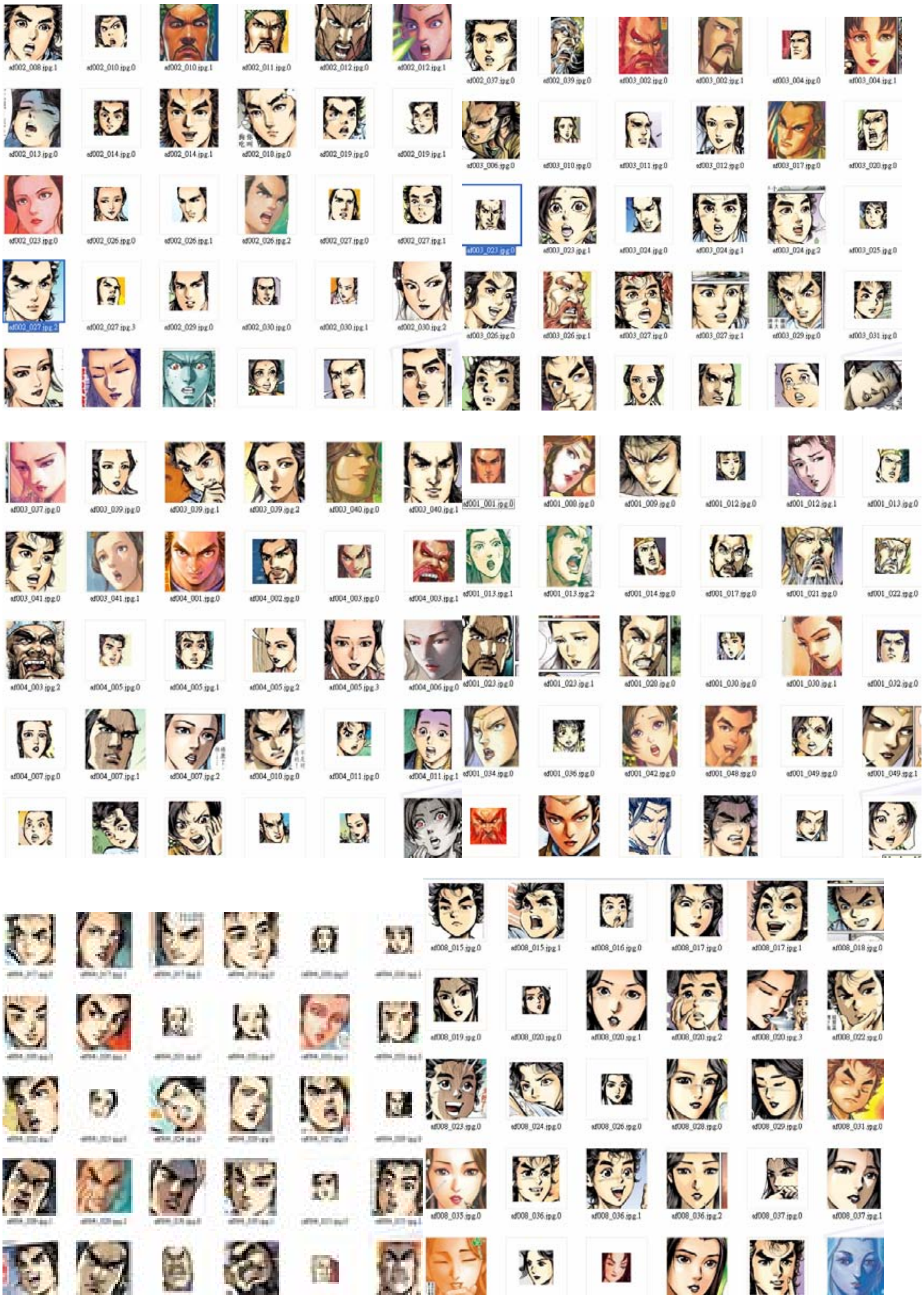
Appendices

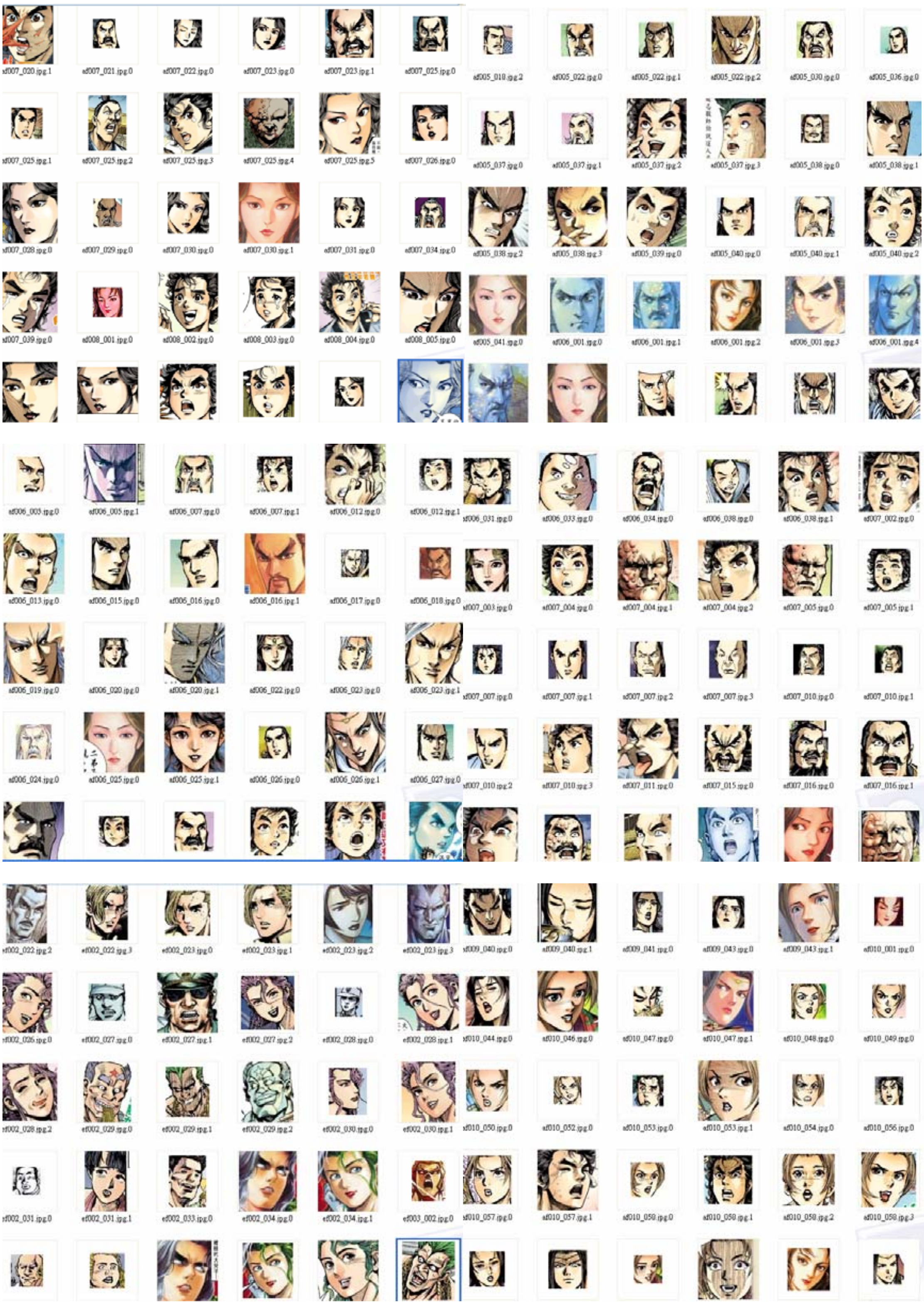
Appendix A -- Monthly Log

Project Month	Progress Log
Sept:	Week 1: Detection Overview Week 2: Color Detection Coding Week 3: Project Plan + improving code Week 4: Comparing Detection results
Oct:	Week 1: Improve Color Detection as to reduce the false detected faces Week 2: Summary of Color Detection phase Week 3: Generate images for Recognition part Week 4: Obtain recognition source code + Recognition overview
Nov	Week 1: Face Recognition Overview Week 2: Study of methods Week 3 +4: Experimenting working methods on comic images (SIFT + EBGM)
Dec	Week 1-4: Experimenting EBGM, Bayesian, LDA and PCA on comic sets

Jan	Coding the application Week 1: building application interface outline window Week 2 + Week 3: Reuniting the ground truth tool and the application Week 4: Add Detection part to the application
Feb	Coding recognition demo
Mar	-Algorithm evaluation, Measurement of effectiveness -Changes to the application system -Add-in functions to the application system -Consolidation of the project
Apr	-Change Request -Reporting

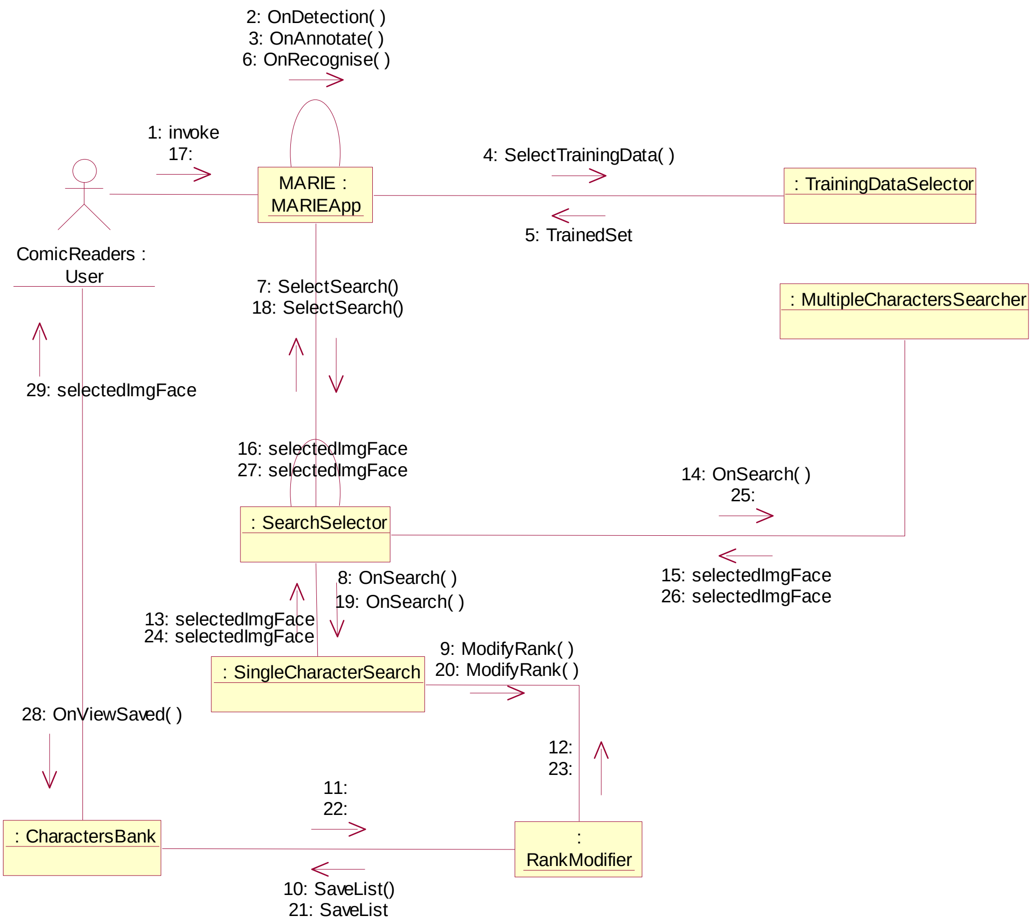
Appendix B – Data Set for Face Recognition



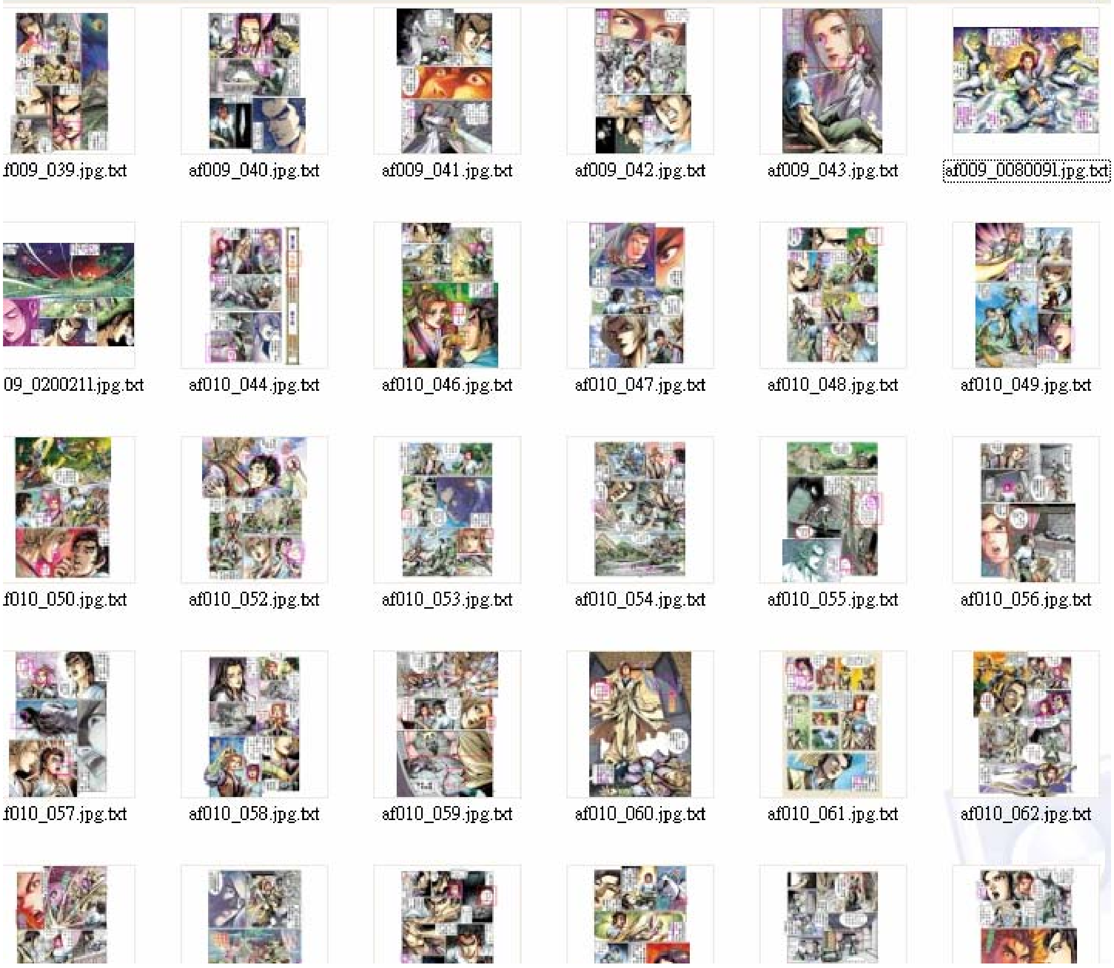


(Not all of them are listed)

Appendix C – Collaboration Diagram of MAIRE



Appendix D – Data Set for Face Detection



(Not all of them are listed)