Progressive Learning Strategy for On-Line Face Recognition and Annotation

Chung-Lun Lu
Dept. of Computer Science
National Tsing Hua University
Hsin-Chu, Taiwan, R.O.C.
chunglunlu@gmail.com

Been-Chian Chien
Dept. of Computer Science and Information Engineering
National University of Tainan
Tainan, Taiwan, R.O.C.
bcchien@mail.nutn.edu.tw

Abstract

On-line human face recognition and annotation need effective image features and efficient adaptive learning methods. In this paper, new features extraction methods are proposed to reduce the number of facial features. Based on these extracted features, a progressive training strategy for adaptive facial learning is applied to generate classifiers for human faces classification. To demonstrate the performance of extracted features, off-line classifiers are used to compare with the progressive strategy. The experimental results show that the extracted 42 facial features can support on-line face recognition for annotating effectively when the progressive training data are collected up to about 40 facial data.

Keywords-facial features extraction; face recognition, Progressive learning, on-line learning

I. INTRODUCTION

Face recognition is an important research problem in computer vision. Many applications are based on high-accurate face recognition technologies. A general definition of face recognitions was given by Zhao et al [20] as follows: identifying the persons from an image whose face image has been recognized and stored in a face database. In general, a generic face recognition system includes three phases: the face detection, the features extraction and the face recognition [20]. The face detection phase first detects and localizes faces from images. Then the features extraction phase extracts some facial features representing the detected face. Finally, the face recognition phase identifies the detected face.

A general face recognition system is quite difficult. It is impossible to recognize any person in an unconstrained circumstance. A face recognizer usually needs to be trained by a large collection of training face images. However, the collection of face images is not easy for some people at a time. It will result in an ineffective face recognizer in such a case. In this paper, we propose a features extraction method that supports progressive training of face recognition for an incremental face images collecting. The proposed method first localizes three areas of facial features, mouth area, nose area, and eyes-eyebrows area. Then 11 feature points are extracted from the localized areas. Finally, the extracted feature points are characterized by 24 normalized distances, 10 eyebrow-position angles and 8 eyebrow shape similarities. We performed three experiments. The first experiment studied the recognition rate using non-boosting learning classifiers. The second and third experiments studied the progressive non-boosting learning techniques and progressive boosting techniques, respectively.

The remaining of this paper is organized as follows: Section 2 reviews the related researches of face recognition and machine learning. Section 3 describes the proposed architecture; the detailed methods and algorithms of the system are discussed in Section 4. The experimental results of face classification are discussed in Section 5. Finally, we summarize the presented work in Section 6.

II. RELATED WORKS

A. *Face Recognition*

The approaches of face recognition are generally divided into three categories: the holistic matching methods, the feature-based matching methods and the hybrid methods [20]. We give a brief summarize of these methods.

- *Holistic matching methods*
  In holistic matching methods, the whole face region is input to a face recognition system. The most famous techniques are Eigenfaces and Fisherfaces.
- *Feature-based matching methods*
  In the feature-based matching methods, local facial features are extracted and then input to a classifier. Instead of extracting the exact locations of features, Nefian et al. [11] extracted the eyes, nose, mouth, and chin by strips of pixels. Wiskott et al. [18] developed a graph matching system which extracted image graph based on the bunch graph for face recognition.
- *Hybrid methods*
  In hybrid methods, both local facial features and the whole face region are used to recognize a face. Pentland et al. [13] proposed a view-based multiple-observer eigenspace technique for face recognition under variable pose. An integration of moment invariant and PCA for face recognition is proposed by Phiasai et al. [14]. Phiasai et al. first used PCA to extract the global feature. If the error of the face

recognition system is unacceptable, then local features will be extracted by moment invariant.

B. Machine Learning

Here we category the classification into non-boosting approaches and boosting approaches. We give a brief summarize of these approaches.

- *Non-boosting*
  In non-boosting learning, the classification result relies on single classifier. Most of native learning models are non-boosting, such as C4.5 [15][16][17], support vector machine [2], hidden Markov model, etc.
- *Boosting*
  Non-boosting learning relies on single classifier. If the single classifier is weak, it will result in high recognition error. To solve this problem, the concept of boosting was posed by Kearns and Valiant in 1988 [10]. The classification result of boosting is voted by a set of classifiers. In 1995, Freund and Schapire proposed AdaBoost (Adaptive Boosting) algorithm [6]. Other boosting approaches are LPBoost (Linear Programming Boosting) [3], LogitBoost [7], BrownBoost [4], etc.

III. THE ARCHITECTURE OF FACE RECOGNITION

The architecture of the proposed face recognition system is shown as Fig. 1. The system consists of three components: the face detector, the facial features extractor, and the face classification. The functions of each component are described as follows:

- *Face Detector*
  The face detector detects and localizes human faces for general images. Each detected face range is segmented as an independent image, called a face image. In this work, the face detector directly used Intel's OpenCV [12] to achieve human face detection and segmentation.
- *Facial Features Extractor*
  The facial features extractor extracts important facial features. In this phase, we develop a feature extraction method which works on color images without wearing glasses. The facial features extractor consists of three subsequent modules: the facial features localization, the extraction of feature points, and the generation of facial features. First, the facial features localization module localizes the facial features, including eyebrows, eyes, nose and mouth, from the face image. Second, the extraction of feature points module extracts 11 feature points from the localized facial features. At last, the generating of facial features module characterizes the 11 extracted feature points by 24 normalized distances, 10 eyebrow-position angles and 8 eyebrow shape similarities.
- *Face classification*
  The face classification component includes classifier and learning algorithm. This component is in charge of face recognition.

Given a new image, the face detector first detects and localizes human faces. Then the detected face image is passed into the facial features extractor, and the 42 facial features are extracted. At last, the extracted facial features are used to train face classifier by learning algorithm for face recognition.

IV. EXTRACTION OF FACIAL FEATURES

This section depicts the features extraction method in detail. This features extractor consists of three subtasks. The first task is to localize mouth, nose, eyes and eyebrows which are known as facial features. The second task is to extract 11 feature points, including two lip corners, one nose tip, two pupils, four ends of eyebrows and two eyebrow tops. The third task is to characterize these extracted feature points by 24 normalized distances, 10 angles, and 8 similarities.

A. Facial Features Localization

To localize facial features, a face image is segmented by a relative positions method, shown as Fig. 2. Let w be the width and h be the height of the face image. The face image is segmented into three areas: the mouth area, the nose area, and the eyes-eyebrows area. Assume that the reference point of the area in the image is the left-top corner. The width of the mouth area is $0.5\times w$ and the height is $0.2\times h$, and the relative position to the reference point is $(0.25\times w, 0.7\times h)$. The width of the nose area is $0.2\times w$ and the height is $0.25\times h$, and the relative position to the reference point is $(0.4\times w, 0.47\times h)$. The width of the eyes-eyebrows area is $0.7\times w$ and the height is $0.3\times h$, and the relative position to the reference point is $(0.15\times w, 0.2\times h)$. An eyebrow area is further segmented from the eyes-eyebrows area in the latter process. Fig. 3 shows the results of segmentation.

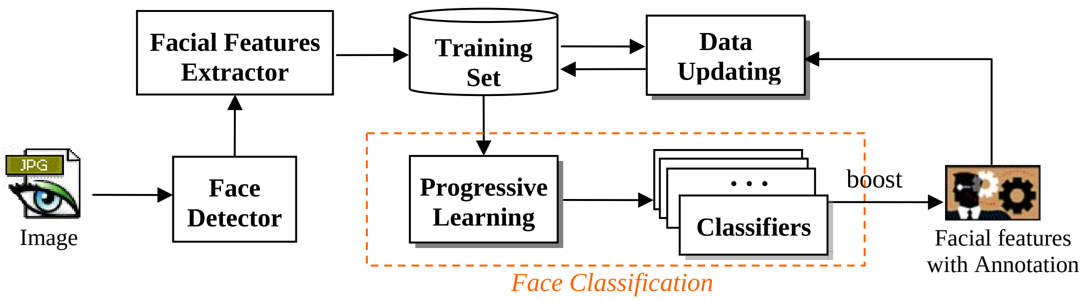


Figure 1. The architecture of the proposed face recognition system.

B. *Extraction of Feature Points*

The extraction module determines 11 feature points for a face image including two lip corners, one nose tip, two pupils, four points of eyebrows ends and two top points of eyebrows, as shown in Fig. 4.

(1) *Lip Corners Extraction*

To find the lip corners, we build a mouth map, *MouthMap*, which is proposed by Hsu et al. [9]. The *MouthMap* is constructed as follows:

$$MouthMap = (C_r)^2 \cdot ((C_r)^2 - \eta \cdot C_r / C_b)^2 . \tag{1}$$

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

where $(C_r)^2$ and C_r/C_b are normalized to [0,255] and *FI* is the face image obtained from face detector. C_b and C_r are pixel values in YC_bC_r color space. As an example image in Fig. 5 (a), Fig. 5 (b) is the image after processing of *MouthMap*. The 10% of the highest pixels in the image of Fig. 5 (b) is then highlighted. The highlighted *MouthMap* is shown in Fig. 5 (c). Then the highlighted *MouthMap* is filtered to reduce noise. After filtering out the highlighted *MouthMap* image, the points of lip corners are determined by taking the most lateral extremes, shown as Fig. 5 (d).

(2) *Nose Point Extraction*

To find the nose tip, we apply a method proposed by Arca et al. [1].

(3) *Pupils Extraction*

To find the pupil points, the *EyeMap*, proposed by Hsu et al. [9], is used. The *EyeMapC* is constructed by

$$EyeMapC = \frac{(C_b)^2 + (\tilde{C}_r)^2 + (C_b / C_r)}{3} . \tag{3}$$

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

The *EyeMapL* is constructed by

$$EyeMapL = \frac{Y(x,y) \oplus s(x,y)}{Y(x,y) \Theta s(x,y) + 1} . \tag{4}$$

where $\oplus$ is dilation operation, Θ is erosion operation, Y is the pixel value in YC_bC_r color space and s is a structuring element.

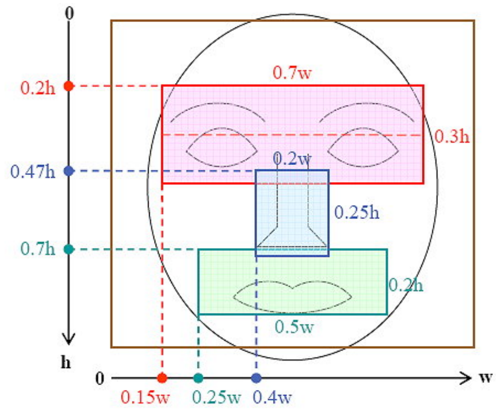


Figure 2. Facial features localization.

Figure 3. Examples of the facial features localization.

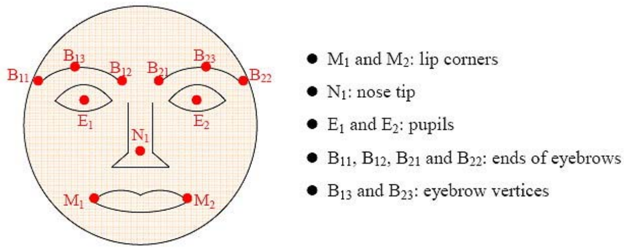


Figure 4. The 11 feature points.

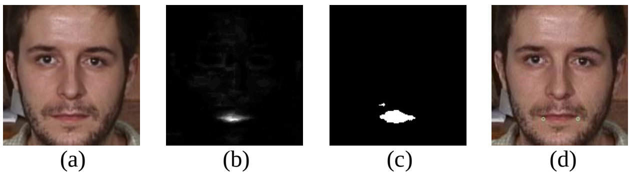


Figure 5. The processing images of lip corners extraction.

Then *EyeMapC* and *EyeMapL* are combined into *EyeMap* by an AND operation,

$$EyeMap = EyeMapC(x,y) \textbf{ AND } EyeMapL(x,y) . \tag{5}$$

The original *EyeMap* is used to detect human faces in images [9]. In this paper, we used *EyeMap* to extract pupils points.We define a new map called *EyeMapD*. The *EyeMapD* is constructed as follows:

$$EyeMapD = EyeMap(x,y) \oplus s(x,y) . \tag{6}$$

To extract the boundary of *EyeMapD*, the *EyeMapD* is first eroded by the structuring element *s*. Then we generate an image, defined as *EyeMapB*. The *EyeMapB* is constructed as follows:

$$EyeMapB = EyeMapD(x, y) - (EyeMapD(x, y) \Theta s(x, y)) \text{.} \quad (7)$$

At last, we separate the *EyeMapB* into two equal parts: the left part and the right part. For each part we find its central point by averaging the coordinate of all white pixels. The central point in the left part is the pupil of left eye; the central point in the right part is the pupil of right eye. The processing images of pupils extraction are shown in Fig. 6. Fig. 6 (a) is the image after *EyeMapC* processing. Fig. 6 (b) is the image after *EyeMapL* processing. Fig. 6 (c) is the image after *EyeMap* processing. Fig. 6 (d) is the image after *EyeMapD* processing. Fig. 6 (e) is the image after *EyeMapB* processing. Fig. 6 (f) shows the extracted result.

(4) Eyebrow Points Extraction

To extract the four points of eyebrows ends and the two top points of eyebrows, we first define an image, called *FeatureImage*. The *FeatureImage* is constructed by the following steps. First, we apply the Prewitt filter to the face image to extract edge. As an example image in Fig. 7 (a), Fig. 7 (b) is the image after performing edge extraction on the face image. The edged image is then contrasted to highlight its edge, shown as Fig. 7 (c). To reduce the noise, we binarize the contrasted image and apply a 5×5 median filter to it. The binarized image is shown in Fig. 7 (d). The *FeatureImage* is obtained after filtering the binarized image, shown as Fig. 7 (e).

After the *FeatureImage* is obtained, we extract the eyebrows points. We first build a binarized image (BI) using Algorithm 1.

<u>Algorithm 1</u>: BI Generation
Input: A face image, *FaceImage*, with resolution $m \times n$ pixels.
Output: A binarized image (BI) with resolution $m \times n$ pixels.
Step 1. Compute the average pixel value, *avg*.
Step 2. For every pixel in *FaceImage*, if *FaceImage*(x, y) >= *avg*, then BI(x, y) = 1. Else BI(x, y) = 0.

Then we combine the binarized image (BI) with *FeatureImage* (FI) using the following algorithm:

<u>Algorithm 2</u>: *EyebrowImage* Generation
Input: FI(resolution: $m \times n$ pixels), BI(resolution: $m \times n$ pixels).
Output: A combined image, *EyebrowImage* (CI), with resolution $m \times n$ pixels.
Step 1. For a pixel in FI, if FI(x, y) is outside the eye and eyebrow area, then CI(x, y) = 0 and go to step 4. Else go to step 2.
Step 2. If FI(x, y) equals to 0 and BI(x, y) equals to 1, then CI(x, y) = 1 and go to step 4. Else go to step 3.
Step 3. CI(x, y) = 0 and go to step 4.
Step 4. Move (x, y) to the next pixel in FI and go to step 1.

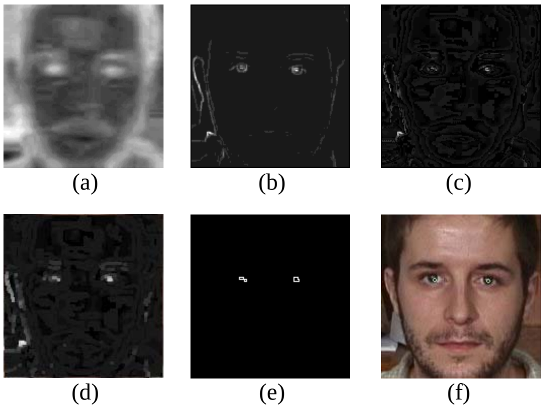


Figure 6. The processing images of pupils extraction.

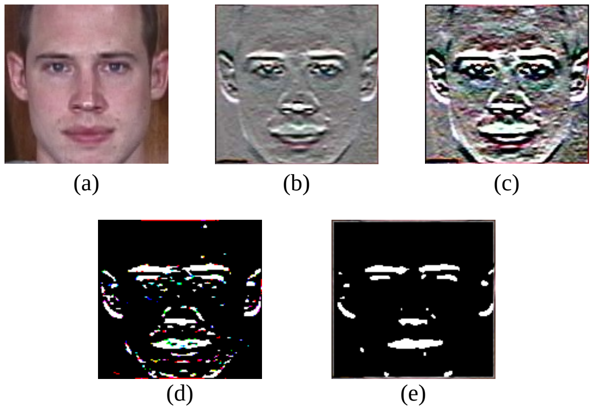


Figure 7. The processing images of building *FeatureImage*.

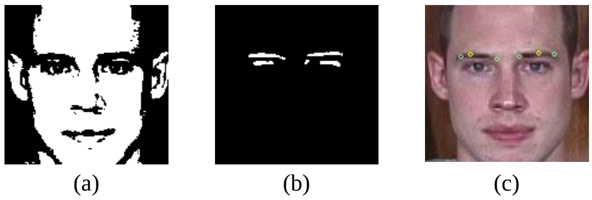


Figure 8. The processing images of eyebrow points extraction.

Finally, the eyebrows ends are determined by taking the most lateral extremes in the image of *EyebrowImage*. And the eyebrows tops are determined by taking the top extremes in the image of *EyebrowImage*. The processing images of eyebrow points extraction are shown in Fig. 8. Fig. 8 (a) is the image after binarizing the face image. Fig. 8 (b) is the image after *EyebrowImage* processing. Fig. 8 (c) shows the extracted result.

C. *Generating of Facial Features*

In this paper, 42 facial features are generated to represent a face image. The 42 facial features include 24 normalized distances, 10 eyebrow-position angles and 8 eyebrow shape similarities. The 24 distances are obtained from the 11 extracted feature points, shown as Fig. 9, and the normalized unit is the distance between the two pupils ($\overline{E_1E_2}$). L1, L11, L12, L21, L22, L23 and L24 are distances between the six eyebrow feature points. L2 is the lip length, the distance between the lip corners. L3, L4, L19 and L20 are distances between the eyebrow area and mouth area. L5 and L6 are distances between the eye area and the mouth area. L7 and L8 are distances between the nose area and the mouth area. L9 and L10 are distances between the eyebrow area and the nose area. L13, L14, L15 and L16 are distances between the eyebrow area and the eye area. L17 and L18 are distances between the eyebrow area and the nose area. The extracted eyebrow points and the normalized unit determine 5 angles for each eyebrow. We also constructed 4 models of eyebrow shapes, shown as Fig. 10. The left eyebrow compares to these eyebrow shape models can obtain 4 similarities; the right eyebrow compares to these eyebrow shape models also can obtain 4 similarities. Fig. 11 shows the extraction results.

V. Experimental Results

To evaluate the performance, we performed three experiments. The experiments referred to two datasets. Dataset 1 is constructed from the Georgia Tech face database (GTDB) [8]. Dataset 1 contains 5 classes and without fixed training and testing sets. There are 10 examples for each class in Dataset1. Dataset 2 is constructed from our own face database (KIDSDB). Dataset 2 contains 2 classes and has fixed training and testing sets. The number of examples of the positive class and the negative class in Dataset 2 are 36 and 20, respectively. The number of training and testing examples of Dataset 2 are 42 and 14, respectively.

A. *Experiment 1: non-boosting learning*

The first experiment studied the recognition rate on different non-boosting learning classifiers. We present results using 16 different learning algorithms on a machine learning tool, Weka [19]. Fig. 12 shows the results tested on Dataset 1; Fig. 13 shows the results tested on Dataset 2. According to the experimental results, we observed that the 10-ford cross validation has the best recognition rate. Also, the accurate extraction of feature points can obtain high recognition rate.

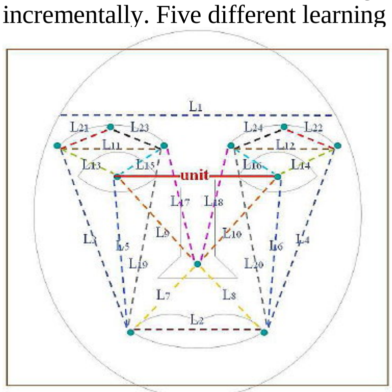
B. *Experiment 2: progressive non-boosting learning*

In the second experiment, we test the performance of the progressive non-boosting learning.

This experiment was tested only on Dataset 2. The testing set is fixed whereas the number of training example is increased incrementally. Five different learning algorithms including C4.5, LibSVM [2], Naïve Bayes, RBF Network, and Random Forest are selected to show the performance of the progressive non-boosting learning. Fig. 14 shows the results.

C. *Experiment 3: progressive boosting learning*

In the third experiment, we test the performance of the progressive boosting learning. This experiment was tested only on Dataset 2. As Experiment 2, the testing set is fixed whereas the number of training example is increased incrementally. Five different learning algorithms including C4.5, LibSVM, Naïve Bayes, RBF Network, and Random Forest are selected to show the performance of the progressive boosting learning. Fig. 15 shows the results.



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

Figure 9. The 24 distances for face characterization

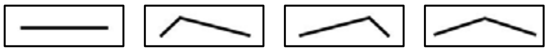


Figure 10. Eyebrow shapes.

Figure 11. Examples of feature points extraction.

According to the results on Experiment 2 and Experiment 3, when the number of training images is up to 30 and 42, an acceptable recognition rates is obtained. The recognition rate increases while the number of training images increases. However, we found that sometimes applying boosting technique may result in worse recognition rates than non-boosting learning. The season of boosting approach is not good as our expectation is because the performance of boosting may be affected by its base model [5].

VI. Conclusion

In this paper, a features extraction method is proposed to overcome the problem that traditional face recognition techniques need the whole training images to be prepared before training. For each training image, we extract 42 facial features, including 24 normalized distance, 10 eyebrow-position angles and 8 eyebrow shape similarities.

We performed three sets of experiments. In the first set of experiments, 87.29% average recognition rate is obtained. The first experiment shows that the proposed features extraction method has high recognition rate on non-boosting learning. In the second and third set of experiments, when the number of training images is up to 30, 85% recognition rate is obtained. So when the number of training images is up to 30, the accuracy of progressive learning will approximate to its batch counterpart (non-progressive learning). Hence, it has high feasibility to apply the proposed features extraction method to progressive training of face recognition for an incremental face images collecting.

REFERENCES

[1] S. Arca, P. Campadelli, and R. Lanzarotti, "A Face Recognition System Based on Automatically Determined Facial Fiducials Points", Pattern Recognition, vol. 39, pp. 432-443, 2006.

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

[3] A. Demiriz, K. Bennett, and J. Shawe-Taylor, "Linear Programming Boosting via Column Generation," Machine Learning, vol. 46, pp. 225-254, 2002.

[4] Y. Freund, "An Adaptive Version of the Boost by Majority Algorithm," Machine Learning, vol. 43, pp. 293-318, 2001.

[5] Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm," Proc. of the 13th International Conference on Machine Learning, pp. 148-156, 1996.

[6] Y. Freund and R. Schapire, "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119-139, 1997.

[7] J. Friedman, T. Hastie and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," Annals of Statistics, vol. 28, no. 2, pp. 337-407, 2000.

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

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

[10] M. Kearns and L. G. Valiant, "Learning Boolean Formulae Or Finite Automata Is As Hard As Factoring," Technical Report TR-14-88, Harvard University Aiken Computation Laboratory, 1988.

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

[12] Open Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/

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

[14] T. Phiasai, S. Arunrungrusmi, and K. Chamnongthai, "Face Recognition System with PCA and Moment Invariant Method," Porc. of the IEEE International Symposium on Circuits and Systems, vol. 2, pp. 165-168, 2001.

[15] J. R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81-106, 1986.

[16] J. R. Quinlan, "Learning Logical Definitions from Relations," Machine Learning, vol. 5, pp. 239-266, 1990.

[17] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.

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

[19] I. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed, Morgan Kaufmann, San Francisco, 2005.

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

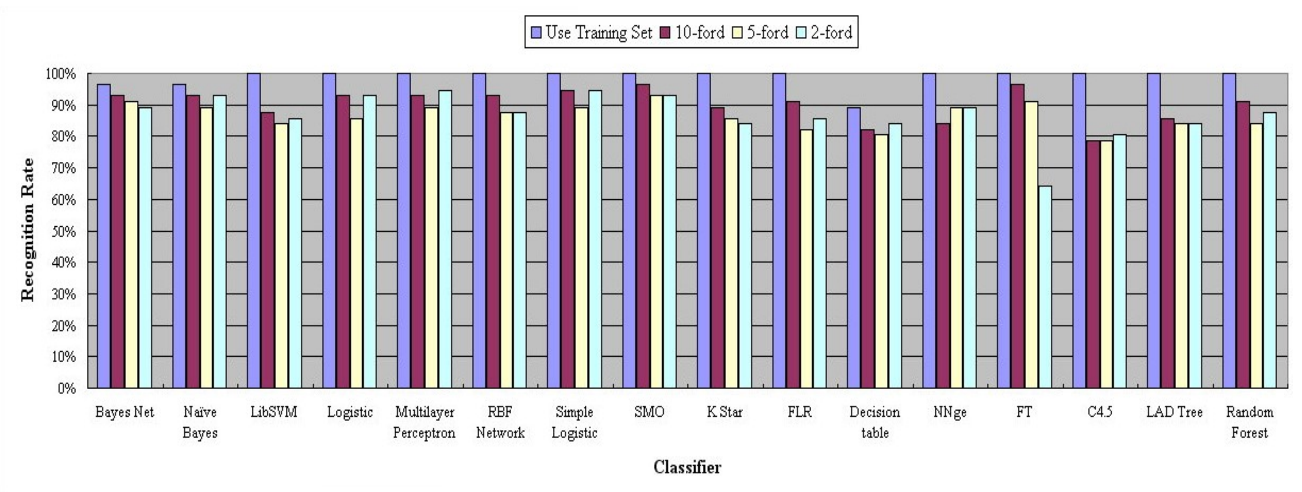


Figure 12. The results of Experiment 1 (Dataset 1).

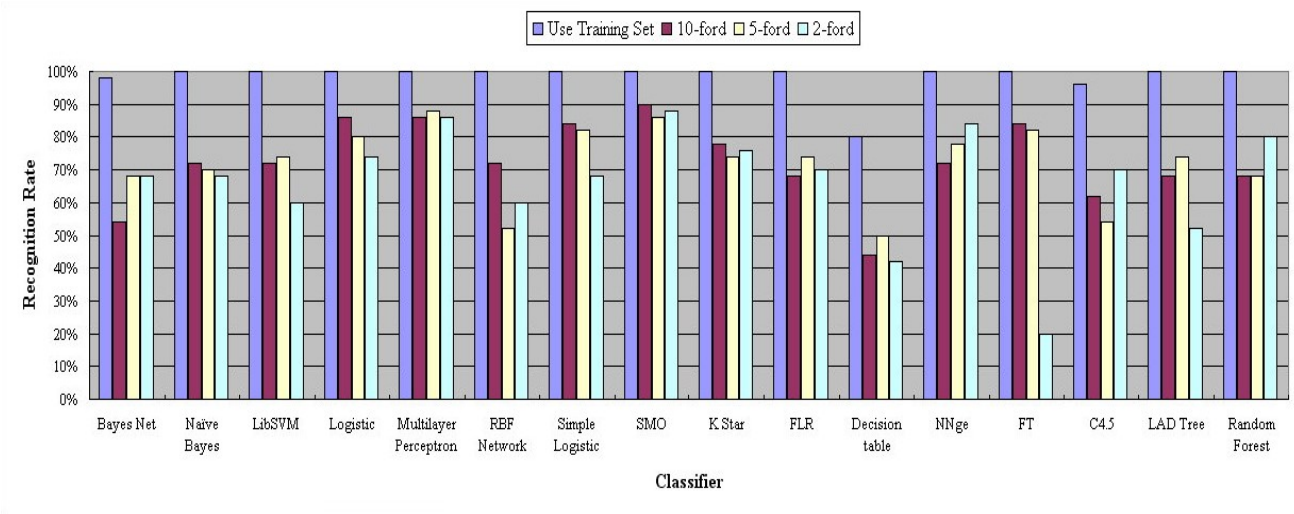


Figure 13. The results of Experiment 1 (Dataset 2).

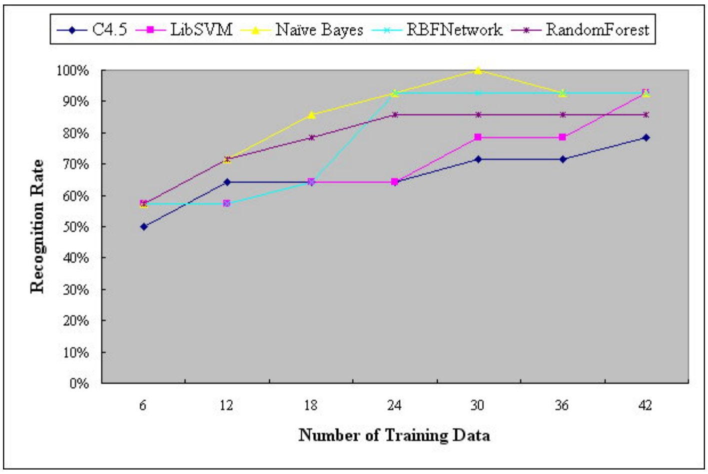


Figure 14. The results of Experiment 2 (Dataset 2).

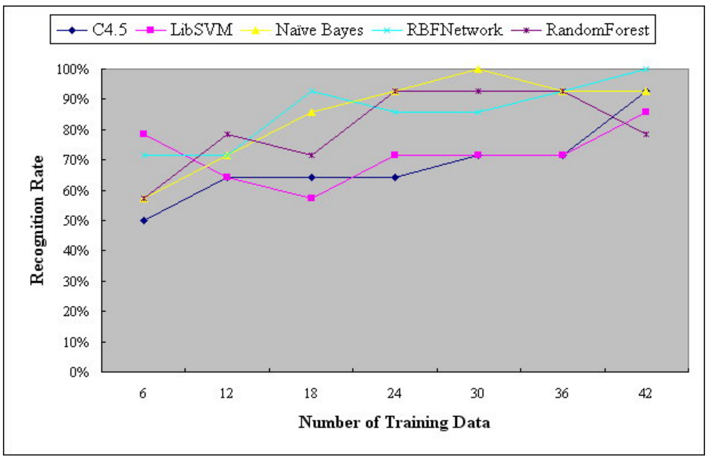


Figure 15. The results of Experiment 3 (Dataset 2).