Pattern Recognition 43 (2010) 1039–1049

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.de/locate/pr

# A data mining approach to face detection

Wen-Kwang Tsao, Anthony J.T. Lee *, Ying-Ho Liu, Ting-Wei Chang, Hsiu-Hui Lin

*Department of Information Management, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan, ROC*

ARTICLE INFO

*Article history:*
Received 28 May 2007
Received in revised form
25 February 2009
Accepted 3 September 2009

*Keywords:*
Face detection
Feature pattern
Maximal frequent pattern
Data mining
Support vector machine

ABSTRACT

In this paper, we propose a novel face detection method based on the MAFIA algorithm. Our proposed method consists of two phases, namely, training and detection. In the training phase, we first apply Sobel's edge detection operator, morphological operator, and thresholding to each training image, and transform it into an edge image. Next, we use the MAFIA algorithm to mine the maximal frequent patterns from those edge images and obtain the positive feature pattern. Similarly, we can obtain the negative feature pattern from the complements of edge images. Based on the feature patterns mined, we construct a face detector to prune non-face candidates. In the detection phase, we apply a sliding window to the testing image in different scales. For each sliding window, if the slide window passes the face detector, it is considered as a human face. The proposed method can automatically find the feature patterns that capture most of facial features. By using the feature patterns to construct a face detector, the proposed method is robust to races, illumination, and facial expressions. The experimental results show that the proposed method has outstanding performance in the MIT-CMU dataset and comparable performance in the BioID dataset in terms of false positive and detection rate.



## 1. Introduction

Face detection is a fundamental problem in many computer vision applications. It can be used to locate a face, and as a front–end for applications such as face recognition system, surveillance and security system, human computer interaction (HCI) system, etc. Face is a highly non-rigid object. The challenge of detecting human faces from an image mostly comes from the variation of human faces such as races, illumination, facial expressions, face scales, head poses (off-plane rotations), face tilting (in-plane rotations), occlusions, etc. Also, environment issues such as lighting conditions, image quality, and cluttered backgrounds may cause great difficulties.

Many face detection methods have been proposed. These methods can be classified into three categories [1]: knowledge-based and feature invariant methods, template matching methods, and appearance-based methods.

The knowledge-based and feature invariant methods locate the features of a face such as eyes, nose, mouth, or even skin color, and then group them together by considering their geometrical relationships [2–4]. Bhuiyan et al. [5] proposed a method to convert pixels in color images from the RGB model to the YIQ model to find skin colors and group them into skin regions. Chiang et al. [3] presented a model to convert pixels from the RGB model to the $r$–$g$ plane and find skin and lip colors in an image between two parabolas on that plane. Once the skin regions are located in an image, facial components or features like eyes, eyebrows, nose or mouth can be detected within these skin regions. A skin region with proper facial components is reported as a face candidate. Lee et al. [6] used the directional template and blob map to locate major facial features, i.e., two eyes and a mouth. Usually, a verification process such as facial components' geometrical relations is needed to reduce false positives [7]. One major problem about knowledge-based and feature invariant methods is that it is hard to translate human knowledge into well-defined rules. If the rules are too strict, they may fail to detect human faces. On the other hand, if the rules are too general, they may result in too many false positives.

Template matching methods aim to find face features that exist even when the pose or lighting conditions are varying [1]. Usually, a standard face template (pattern) is manually predefined or constructed by a function. The existence of a face is determined based on the similarity or correlation between the input image and the standard face pattern in terms of eyes, nose, mouth, and face contour. Silhouettes have also been used as templates for locating face candidates [8]. Bhuiyan et al. [5] formed the template of a face image, which is obtained from averaging the gradation levels of pixels of face samples. The template face image is then shifted through the whole image to find the location with the most suitable matches. Jesorsky et al. [9] used a face

* Corresponding author.
*E-mail addresses:* d93725001@ntu.edu.tw (W.-K. Tsao), jtlee@ntu.edu.tw (A.J. Lee), d94725011@ntu.edu.tw (Y.-H. Liu), r94725006@ntu.edu.tw (T.-W. Chang), d8725003@ntu.edu.tw (H.-H. Lin).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.09.005

template to roughly locate a face candidate. Then an eye template is exploited to verify the located face candidates. One disadvantage of template matching methods is that it is not easy to obtain a good template from training face images. Another disadvantage is that it is difficult to enumerate templates for different poses.

In contrast to template matching methods, appearance-based methods learn templates from training samples (faces and non-faces) instead of learning from experts; therefore the training process is usually time-consuming. However the detection speed of these methods is usually fast, sometimes even real-time. This category of methods includes statistical methods [10], neural networks [11–13], support vector machines (SVM) [14–16], and multiple classifier combination [17–19].

Shih and Liu [15] proposed the DFA-SVM method which integrates the discriminating feature analysis, distribution-based face class modeling, and SVM. Heisele et al. [16] built a 5-level hierarchy of SVM classifiers with lower level classifiers rejecting most parts of the background, and the upper level classifiers performed the detailed detection. Then, PCA is applied to the top-level classifier to choose relevant image features. With the combination of a hierarchy of SVM classifiers and feature reduction, the detection process is speeded up in comparison with similar classification methods [20].

Viola and Jones [17] proposed an object detection scheme, which is fast enough for real-time applications and can be effectively applied to face detection. This is accomplished by the integration of a new image representation called integral image, a learning algorithm based on AdaBoost [21], and a cascade scheme is to combine the boosted classifiers. Roth et al. [22] proposed a method, called SNoW, for detecting faces with different features, expressions, and poses under different lighting conditions. Nilsson et al. [23] proposed an extended version of the SNoW classifier by a split-up process. One of the drawbacks of the SNoW-based methods is that they require a large number of face and non-face patches. Cristinacce and Cootes [24] used a joint shape and texture appearance model to generate a set of region template detectors, where a face is detected by aggregating the responses of these detectors.

One problem of the appearance-based methods is that it needs a lot of positive and negative training examples, and the learning process is very time-consuming. Another problem is that the learned result is hard to interpret, thus is difficult to be adjusted to fit various applications.

Of the three categories of methods mentioned above, the disadvantages of these methods are (1) the knowledge and the relation of the features is hard to define, (2) the training process of the appearance-based methods is time-consuming, or (3) a good template is hard to obtain.

Therefore, in this paper, we propose a data mining approach to obtain the features patterns of human faces automatically and efficiently. We first apply Sobel's edge detection operator, morphological operator, and thresholding to each training image, and transform it into an edge image. Next, we use the MAFIA [25] algorithm to mine the maximal frequent patterns from those edge images and obtain the positive feature pattern. Similarly, we can obtain the negative feature pattern from the non-edge images, each of which is a complement of an edge image. Based on the positive and negative feature patterns mined, we construct a face detector containing three cascaded classifiers to prune non-face candidates. That is, we not only focus on the traditional facial feature patterns like eyes or mouth, which are called positive feature pattern, but also focus on the facial regions that do not contain any facial feature patterns like cheeks, which are called negative feature pattern.

Unlike knowledge-based and feature invariant methods, our proposed method can find the feature patterns automatically. Moreover, by using both the positive and negative feature patterns to construct a face detector, our proposed method is robust to races, illumination, and facial expressions.

The rest of the paper is organized as follows. The preliminary concept and problem definitions are described in Section 2. Our proposed method is presented in detail in Section 3. The performance analysis is shown in Section 4. Finally, the conclusions and future work are discussed in Section 5.

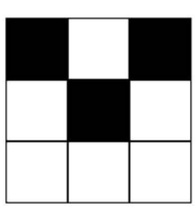


**Fig. 1.** An example image.

## 2. Preliminary concept and problem definition

Before describing our proposed method, we define some notations used later.

*Window*: A window is a fix-sized region in an image.

*Item*: An item is a pixel in a window. An item is denoted by its coordinate $(x,y)$. That is, the item is located at the $x$th column and the $y$th row, where the column and row numbers are started from 0.

*Pattern*: A pattern in an image contains a set of items, which contains all items' coordinates. For example, a pattern in the image shown in Fig. 1 can be presented as $\{(0,0), (2,0), (1,1)\}$.

*Super-pattern*: If every item in pattern $X$ can be found in pattern $Y$, we can say that $Y$ is a super-pattern of $X$ or $X$ is a sub-pattern of $Y$.

*Support*: The support of a pattern is defined as the percentage of images containing the pattern in the database.

*Frequent pattern* (*FP*): A pattern is *frequent* if its support is not less than the user-specified minimum support threshold.

*Maximal frequent pattern* (*MFP*): A frequent pattern $X$ is *maximal* if none of $X$'s super-patterns are frequent.

## 3. Our proposed approach

Our proposed approach contains two phases: training phase and detection phase.


**Fig. 2.** Processed image: (a) input image; (b) after histogram equalization; (c) after horizontal edge detection; (d) filtered edge; (e) after dilation.

### 3.1. Training phase

The training phase consists of three stages: (1) preprocessing, in which we use Sobel's edge detection operator to extract the edges for each training image, which is called an edge image; (2) finding the positive and negative feature patterns, in which we use the MAFIA algorithm [25] to get the maximal frequent patterns from the edge images obtained from the first stage. Among the maximal frequent patterns mined, we select one pattern with most facial features. The selected pattern is called positive feature pattern. Moreover, we use the MAFIA algorithm to get the maximal frequent patterns from the non-edge images, each of which is a complement of an edge image. Similarly, we can obtain the negative feature pattern; (3) constructing the face detector, in which we use the positive and negative feature patterns mined in the second stage to construct a face detector containing three cascaded classifiers, namely, variance, face feature, and kd-tree-based support vector machine (SVM).

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

**Fig. 3.** Sobel's horizontal edge detector.

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**Fig. 4.** The structuring element of the dilation operator.

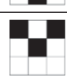
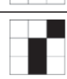
| Image ID | Image | Patterns |
|---|---|---|
| I1 |  | {(0,0), (2,0), (1,1), (1,2)} |
| I2 |  | {(0,0), (2,0), (1,1)} |
| I3 |  | {(2,0), (1,1), (1,2)} |

**Fig. 5.** An example training database.

#### 3.1.1. Preprocessing

In this stage, we extract the key components of a human face. Fig. 2(a) is the input gray-level image. First, we use the histogram-equalized method [26] to the input image and obtain the image as shown in Fig. 2(b). We then apply Sobel's $3 \times 3$ operators to extract the edge image for the input image. Because facial components are mostly composed of horizontal lines, we only adopt a $3 \times 3$ horizontal operator as shown in Fig. 3. After applying Sobel's operator to the image shown in Fig. 2(b), we can obtain a gray-level image as shown in Fig. 2(c), where contours are in bright pixels. Then, we use Eq. (1) to filter out the noisy pixels. The resultant image is shown in Fig. 2(d).

$$f(i) = \begin{cases} 255 & \text{if } i > \mu + c\sigma \\ 0 & \text{others} \end{cases} \tag{1}$$

where $i$ is the intensity of an input pixel, $\mu$ is the average intensity of the image, $c$ is a constant, and $\sigma$ is the standard deviation of intensities of the image.

The last step is to apply the dilation operator around the filtered contour. The structuring element of the dilation is shown in Fig. 4, and the resultant image is shown in Fig. 2(e).

#### 3.1.2. Finding the positive and negative feature patterns

Next, we use the MAFIA algorithm [25] to mine maximal frequent patterns from the edge images obtained from the previous stage. The MAFIA algorithm can efficiently generate maximal frequent patterns from a database. The idea behind this is that several feature pixels tend to co-occur in a human face and those feature pixels will form a maximal frequent pattern.

Because we do not need all of the maximal frequent patterns to detect a face, we select one that has most facial features. The

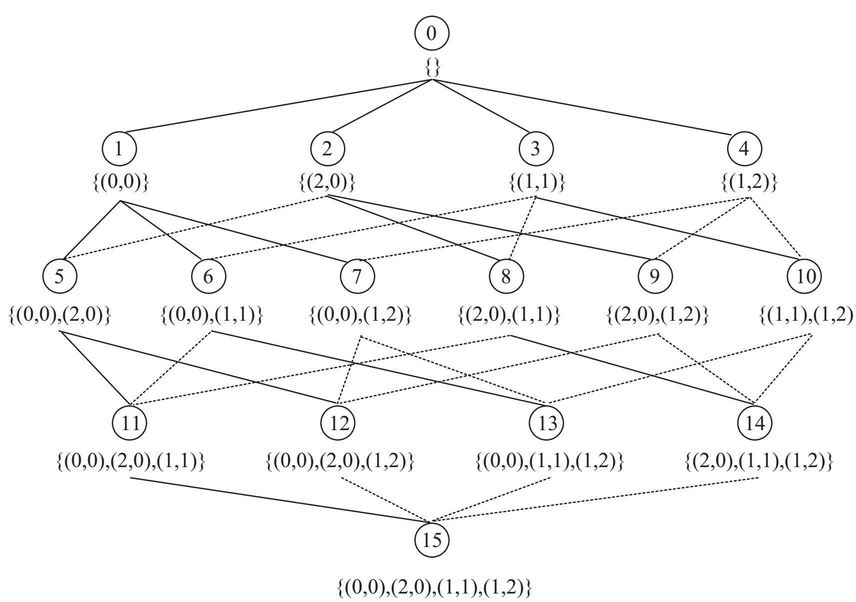



**Fig. 6.** The lattice for four items.

maximal frequent pattern selected is called the positive feature pattern. Then, we apply the MAFIA algorithm on the non-edge images, each of which is a complement of an edge image. Similarly, we can obtain the negative feature pattern.

The MAFIA algorithm is used to find maximal frequent patterns among all training edge images. We demonstrate how it works by the following example. Fig. 5 shows an example training database containing three edge images of size $3 \times 3$. Let the minimum support threshold be 66%. There are four distinct items in total: (0,0), (2,0), (1,1), and (1,2). The lattice for these four items is shown in Fig. 6, where only the solid edges need to be traversed by the MAFIA algorithm.

The MAFIA algorithm traverses the lattice in a depth-first search manner. When it traverses a node, it checks the node's support and prunes away the nodes whose patterns are infrequent or the nodes that cannot be maximal frequent patterns. We start from node 0, node 1, node 5, node 11, and stop at node 15 since node 15 is the deepest node in the left-most branch. We find the pattern {(0,0), (2,0), (1,1), (1,2)} of node 15 is infrequent since its support is 0.33. Next, we go back to node 11. We count the support of the pattern of node 11, and obtain its support=0.66. Thus, {(0,0), (2,0), (1,1)} is frequent. Because the pattern of node 11 is frequent, the patterns of its ancestors are the sub-patterns of {(0,0), (2,0), (1,1)}. That is, nodes 5, 6, 8, 1, 2, 3, and 0 can be pruned away. We continue to traverse nodes 12, 13 and 7. However, the supports of {(0,0), (2,0), (1,2)} (node 12), {(0,0), (1,1), (1,2)} (node 13) and {(0,0), (1,2)} (node 7) are 0.33, so they are infrequent. Next, node 14 is traversed. The support of {(2,0), (1,1), (1,2)} is 0.66, so it is frequent. Consequently, nodes 9, 10, and 4 are pruned away since the patterns of these nodes are the sub-patterns of {(2,0), (1,1), (1,2)}. Finally, we obtain two maximal frequent patterns, which are {(0,0), (2,0), (1,1)} (node 11) and {(2,0), (1,1), (1,2)} (node 14), as shown in Fig. 7. The maximal frequent patterns obtained in the training phase are recorded in the form of $\{i_1, i_2, i_3, \ldots, i_n\}$ [support], where $i_j$ is an coordinate within the edge image, $1 \le j \le n$.

Some maximal frequent patterns obtained from 100 sample images are shown in Fig. 8, where each row represents a maximal frequent pattern, and the number in the brackets is the support of the maximal frequent pattern. Fig. 9 shows five maximal frequent patterns mined from the edge images, where the white pixels in each image are located approximately in the regions of eyes, eyebrows, nose, or mouth. Fig. 10 shows five maximal frequent patterns mined from the non-edge images, where the gray pixels in each image are located approximately in the regions of cheeks. We select one maximal frequent pattern mined from the edge images that has most facial features as the positive feature pattern, as shown in Fig. 9(c). Similarly, we can obtain the negative feature pattern, as shown in Fig. 10(c). Fig. 11 shows the merged result of the selected positive and negative maximal frequent patterns.

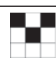
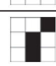
| | | |
|---|---|---|
| Maximal frequent pattern 1 |  | {(0,0), (2,0), (1,1)} [0.66] |
| Maximal frequent pattern 2 |  | {(2,0), (1,1), (1,2)} [0.66] |

**Fig. 7.** The maximal frequent patterns found.

{(15,4), (4,6), (5,6), (6,6), (7,6), (13,6), (14,6), (3,7), (4,7), (5,7), (6,7), (7,7), (13,7), (14,7), (15,7), (16,7), (8,12), (9,12), (10,12), (11,12)} [0.95]
{(4,6), (6,6), (7,6), (13,6), (14,6), (3,7), (4,7), (5,7), (6,7), (7,7), (13,7), (14,7), (15,7), (16,7), (8,12), (9,12), (10,12), (11,12), (7,15)} [0.95]
…

**Fig. 8.** An example of maximal frequent patterns.

### 3.1.3. Constructing the face detector

Based on the positive and negative feature patterns mined, we adopt the coarse-to-fine and simple-to-complex strategy to construct the face detector, which contains three classifiers as shown in Fig. 12. First, the variance classifier is used for pruning windows with a small variance. Next, the face feature classifier is used to select the windows with most facial features. Finally, a kd-tree-based SVM is used for refining the final results.

A threshold learning process is required for the first two classifiers. We input all the faces in the training database to each

**Fig. 11.** The merged result.

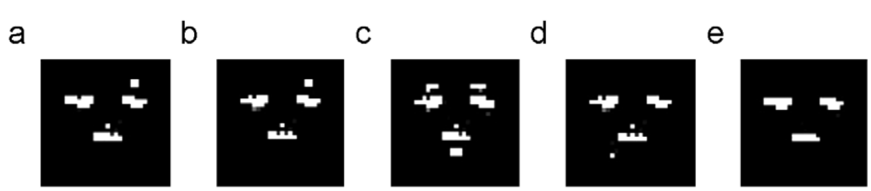



**Fig. 9.** Maximal frequent patterns mined from edge images.


**Fig. 10.** Maximal frequent patterns mined from non-edge images.

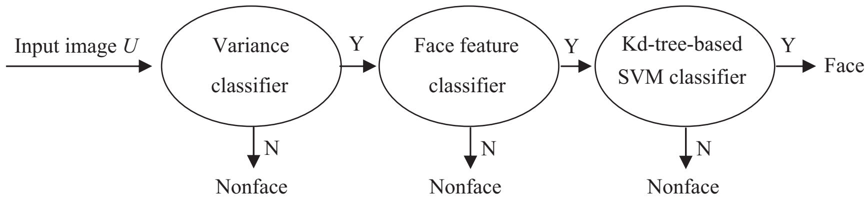



**Fig. 12.** The face detector.

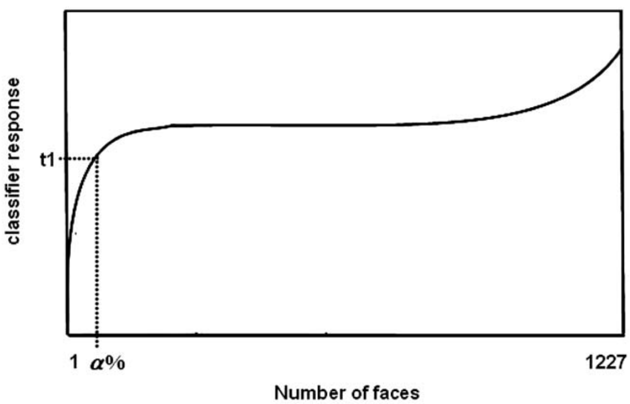



**Fig. 13.** Threshold learning.

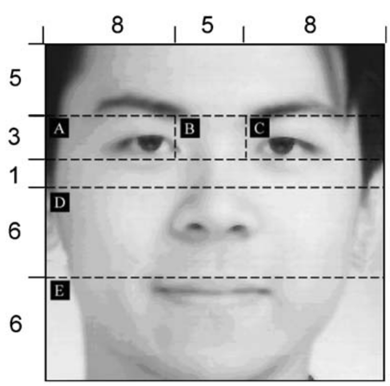



**Fig. 14.** The labeled regions in a 21 × 21 window.

classifier and select classifier threshold based on the responses as shown in Fig. 13, where the selected threshold can let $(100-\alpha)\%$ of faces pass. The experimental results show that we have an acceptable detection rate on faces and an acceptable rejection rate on non-faces when $\alpha=5$.

*Variance classifier*: In order to improve the computational efficiency, we apply an early-exclusion criterion [15] to eliminate the windows to be compared. This criterion divides a window into five regions as shown in Fig. 14, where the numbers beside the figure are distances in terms of pixels. The regions are the left eye ($A$), the nose bridge ($B$), the right eye ($C$), the nose ($D$), and the mouth ($E$). We first calculate the variances in regions $D$ and $E$, respectively, and exclude the window if either variance is smaller than the predefined threshold. If the window passes the first step, we calculate the values $m_a$, $m_b$, and $m_c$ in regions $A$, $B$, and $C$, respectively. $m_a$ is the average intensity of those pixels that are darker than the average intensity in region $A$, $m_b$ is the average intensity of those pixels that are brighter than the average intensity in region $B$, and $m_c$ is the average intensity of those pixels that are darker than the average intensity in region $C$. We exclude the remaining windows if $m_b < k^*m_a$ or $m_b < k^*m_c$, where $k$ is a controlling factor.

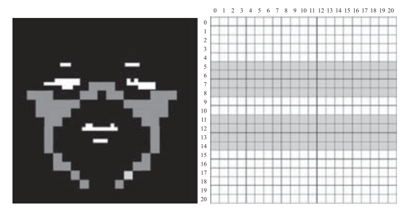


**Fig. 15.** The rows used to form a feature vector.

*Face feature classifier*: The face feature classifier utilizes the positive and negative feature patterns mined to select the images with the most facial features and discard the non-face images. To simplify the process, we compute the difference between the summations of pixel intensities between the positive and negative feature patterns, on both raw and edge images, i.e. Fig. 2(a) and (c).

$C_1$: summation of pixel intensities of the positive feature pattern in the raw image.
$C_2$: summation of pixel intensities of the negative feature pattern in the raw image.
$C_3$: summation of pixel intensities of the positive feature pattern in the edge image.
$C_4$: summation of pixel intensities of the negative feature pattern in the edge image.
Rule 1: $C_1-C_2>T_1$,
Rule 2: $C_3-C_4>T_2$,
Rule 3: $T'_{i,lower} < C_i < T'_{i,upper}$, $i=1, 2, 3, 4$,

where the thresholds $T_1$, $T_2$, $T'_{i,lower}$ and $T'_{i,upper}$, are decided by the threshold learning process shown in Fig. 13. Any image satisfying rules 1, 2 and 3 will be considered as a candidate and passed to the next classifier.

*kd-tree-based SVM classifier*: In order to increase the accuracy of the face detector and avoid a sharp increase in false positives, we employ a kd-tree-based SVM classifier as at the last stage of the face detector, where all true and false positives obtained from the first two classifiers are input to the classifier as the training data.

In this stage, we select eight rows of a raw image to form a feature vector as shown in Fig. 15. That is, the fifth to eighth and the eleventh to fourteenth rows are selected. The feature vector contains the pixel intensities and the coefficients of the HAAR transform of these eight rows. Since each row contains 21 pixels, the feature vector has $21*8*2=336$ dimensions.

The SVM is a widely known approach of learning from training examples [14–16,29,33]. For a binary classification problem, where a dataset $D=\{(x_i,y_i)\}_{i=1}^{l}$ contains $l$ feature vectors and $y_i \in \{-1,1\}$, the SVM classifies the vectors into two classes with the least error. Intuitively, we are looking for a hyperplane so that the margin between both classes is maximized, where the margin is defined as the sum of the distances of the hyperplane from the closest vector of both classes. An optimal hyperplane used to classify the dataset into two classes is shown in Fig. 16, where $v_1$, $v_2$, $v_3$ are the support vectors and lie at the border between both classes.

In most cases, it may be hard to find a hyperplane to classify every vector in the dataset into a correct class. But we can find a hyperplane that maximizes the margin and minimizes the number of mis-classified vectors. If the dataset cannot be well classified by a linear classifier (a hyperplane), we can transform the vectors in the dataset into a higher dimensional space by a kernel function. However, the kernel evaluation in a SVM classifier is very time-consuming. To speed up the detection process, we adopt the kd-tree [28] to split the feature space into several subspaces before training the SVM.

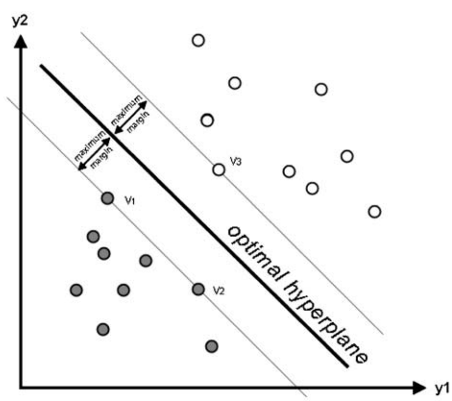



**Fig. 16.** A binary classification problem.

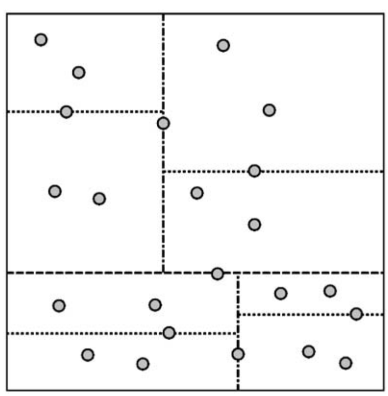


**Fig. 17.** A kd-tree example.

The kd-tree [28] is abbreviated from $k$-dimensional binary search tree, which is a natural generalization of the standard one-dimensional binary search tree. In the tree building process, we first project all feature vectors in a space onto an axis of the feature space and then choose the median of the projected feature vectors to divide the space into two subspaces so that each divided subspace contains equal number of feature vectors. The above step is recursively performed until each divided subspace contains no more than $\beta$ feature vectors. Fig. 17 demonstrates how the kd-tree works, where the dataset contains 23 feature vectors and $\beta=3$. We first project all the feature vectors onto the $y$-axis and choose the median to divide the feature space into two subspaces. For each subspace, we project all the feature vectors in the subspace onto the $x$-axis and choose the median to divide the subspace into two subspaces, each of which is further divided into two subspaces. Finally, we divide the feature space into 8 subspaces, each of which contains no more than 3 feature vectors.

In the experiments, we set $\beta=2000$. For the feature vectors in each divided subspace, we apply a SVM [15,29] to classify those feature vectors into face and non-face classes.

### 3.2. Detection phase

In the detection phase, a sliding window approach is used to search for the faces in a testing image. For each sliding window, we extract the regions of size $21*21$ from every location in the input image pyramid by downsampling. The window is then preprocessed by using the steps in the first stage of the training phase as shown in Fig. 18.

After a window is preprocessed, it is input to the three classifiers shown in Fig. 12. If the window passes these three classifiers, we collect it into the candidate pool and compute its rank as $(C_1+C_3)-(C_2+C_4)$. Otherwise, it will be dropped immediately. To avoid multiple hits on a single face, we merge the overlapping windows of the same size in the candidate pool into one by comparing the ranks among them so that the window with the highest rank will be kept. A window overlaps the other window if the distance between the centers of both windows is less than one fifth of the window size.

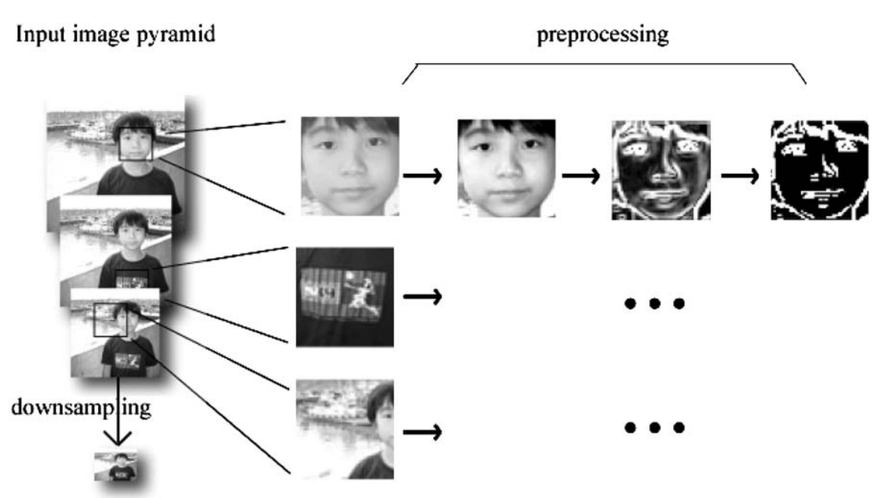



**Fig. 18.** The preprocessing.

**Fig. 19.** Example frontal images used to mine feature patterns.

### 3.3. Discussion

Human's faces have a lot of variations in poses, expressions, lighting conditions, etc. There exist some frequent patterns among face images, and these patterns can be used to distinguish faces from non-faces. The proposed method first finds the positive feature pattern from the edge images and the negative feature pattern from the non-edge images. By using both positive and negative feature patterns, we capture most dominant features of human faces. Based on the positive and negative feature patterns mined, we adopt the coarse-to-fine and simple-to-complex strategy to construct the face detector. When a testing window is compared with the positive and negative feature patterns using face feature responses, slight variations in poses and facial expressions can be tolerated.

Some regions of a face have higher impact on the face detection. Based on the positive and negative feature patterns, we form a feature vector, which has fewer dimensions than that used in the DFA-SVM method [15]. By reducing the number of dimensions of feature vectors, the training and detection processes can be speeded up.

Furthermore, the feature patterns obtained are interpretable. We can relate each part of the feature patterns to the facial components like eyebrows, eyes, nose, mouth, or cheeks. Schneiderman and Kanade [10] argued that the intensity patterns around the eyes are much more important than others. Rowley et al. [11] also pointed out that their network relied mostly on the eyes, and then on the nose and the mouth. The other studies [15,23,29] attempted to locate the dominant region by using an oval mask for ignoring background pixels in four corners of a face image. Instead of labeling these regions manually, our proposed method can automatically find them by using the positive and negative feature patterns mined. Since the patterns can be obtained directly from the training data, no prior knowledge is required to find it.

## 4. Performance analysis

In this section, we conducted the experiments to evaluate the performance of the proposed method, the SNoW-based method [23], the MSRA method [29], the Rowley–Baluja–Kanade method [11], the Viola–Jones method [32], and the New Jersy DFA-SVM method [15]. The MAFIA program was obtained from Himalaya data mining tools [27]. The MSRA and DFA-SVM methods, Viola–Jones method and our proposed method were implemented by using C++ of Visual Studio 2005. The source code of the Rowley–Baluja–Kanade method was downloaded from their website [34]. The SNoW-based method [22] provided the ROC curves of both MIT-CMU and BioID datasets; however, the ROC curve of the MIT-CMU dataset was based on 130 images. Thus, we used the binary file, obtained from the authors [35], to generate a point of the ROC curve for the MIT-CMU dataset with 125 images.

All experiments were performed on an IBM compatible PC with an Intel Core 2 Duo 6550 Dual 2.33 GHz, 2 Gb main memory running on Microsoft Windows XP Professional.

There are two training datasets used in the experiments. One consists of 100 frontal portrait images, which are used for mining positive and negative feature patterns. All of them are students from National Taiwan University. The 58 of them are male, and 42 are female. Some example images of the first dataset are shown in Fig. 19. The other contains 1000 images, which are used for learning the thresholds of the first two classifiers and training the kd-tree-based SVM. Some example images of the second dataset are shown in Fig. 20. This dataset contains 1227 faces including frontal faces, line-drawn faces, partially occluded faces, and cartoon faces, which are selected from our own database. In the first two classifiers, we use the 1227 faces for threshold learning. The true and false positives obtained from the first two classifiers are used to train the third classifier, kd-tree-based SVM. There are two testing datasets for performance evaluation. One is the MIT-CMU dataset [30], and the other is the BioID dataset [31]. They both are public available and used in many previous studies [11,15,17,23,29]. From the MIT-CMU dataset, we select 125 images containing 483 faces that do not include the manually drawn faces. The BioID dataset contains 1520 images with 1521 faces.

**Fig. 20.** Example images for learning thresholds and training the SVM.

In the detection phase, we apply a $21 \times 21$ window to slide across a testing image in different scales. The window is shifted by 2 pixels in the $x$ (or $y$) direction each time. We scale the image by a factor of 1.25 until the image is smaller than $21 \times 21$.

### 4.1. Mining feature patterns

The training images are of size $21 \times 21$ pixels. The minimum support of the maximal frequent patterns mined from the edge images in the training database is set to 0.95, while the minimum support of the maximal frequent patterns mined from the non-edge images is set to 0.98. After running the MAFIA algorithm, we obtain 77 positive and 60 negative maximal frequent patterns. It only takes less than 1.5 s to generate 77 maximal frequent patterns from the edge images and less than 2 s to generate 60 maximal frequent patterns from the non-edge images.

### 4.2. Experimental results

The experimental results show that, although the proposed method addresses the detection of frontal faces, the MIT-CMU images that have large pose-angled faces are still correctly detected. Fig. 21 shows that the proposed method can detect multiple faces in an image, including faces with different scales, slightly rotated faces, and faces in low-quality images.

**Fig. 21.** Examples of detecting faces in the MIT-CMU dataset.

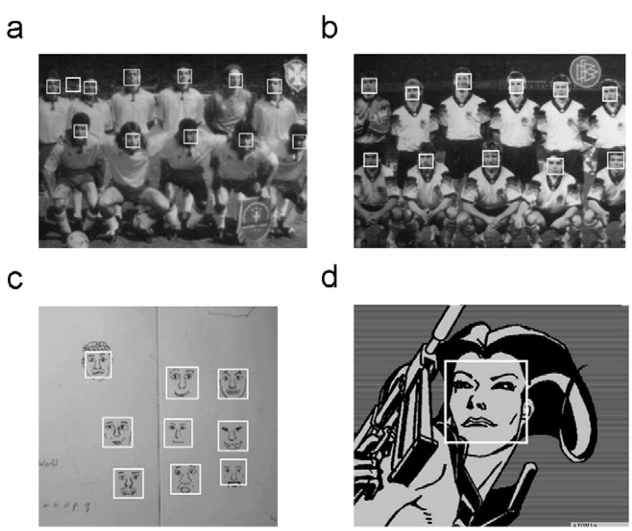



**Fig. 22.** The faces with cluttered backgrounds and the hand-drawn faces.

The faces in both datasets have different sizes, poses, expressions, and lighting conditions, but the proposed method can handle them well. The reason our proposed method works well in those datasets is that the proposed method utilizes the most dominant features obtained from the feature pattern mined. The feature patterns can be frequently found in face images, which do not change much in different categories or sources of testing images. Thus, the feature patterns can be used to pick faces from testing images. Moreover, the k-d-tree SVM classifier shows its feasibility to prune the non-face windows and generate a small number of false positives. The proposed method can successfully detect the faces from cluttered backgrounds as shown in Fig. 22(a) and (b). It can also detect the hand-drawn faces as shown in Fig. 22(c) and (d). The occluded faces can also be successfully detected by the proposed method as shown in Fig. 23. These faces are partially occluded but still have face feature patterns available.

The aliens, who have face feature patterns, can also be detected by the proposed method as shown in Fig. 24. However, when the illumination is too dark and the feature patterns are not clear, the proposed method may fail to detect those faces. In Fig. 24(c), the faces of Guinan and Worf are missed by the proposed method. Since Geordi La Forge wears the VISOR and some face features are lost in his face, the proposed method cannot detect the face. A false positive in the BioID dataset is shown in Fig. 25(a), where the man has a moustache. A missing face in the MIT-CMU dataset is shown in Fig. 25(b), where the man wears glasses.

The proposed algorithm is fast enough for a real-time face detector. We utilize the integral image approach to compute the variance and sum up the intensity values, apply the feature patterns mined to form the feature vectors, and use the k-d-tree to speed up the SVM. From the experiments, the proposed method takes less than 300 ms for scanning 43 025 windows in a $384 * 286$ image (the size of an image in the BioID dataset).

### 4.3. Performance comparisons

The ROC curves of our proposed method and the comparing methods are shown in Figs. 26 and 27. The images in the BioID dataset are all of the same size and do not have many variations in scales, facial expressions, poses, and lighting conditions. In comparison to the MIT-CMU dataset, all the methods have higher detection rates and fewer false positives in the BioID dataset. We observe that the order of the ROC curves of those methods might not be the same in these two datasets. For example, the proposed method performs the best in the MIT-CMU dataset while the SNoW method performs the best in the BioID dataset. It is because each method has its pros and cons in different testing images. For example, the SNoW method has nearly 100% detection rate in the BioID dataset, and the MSRA

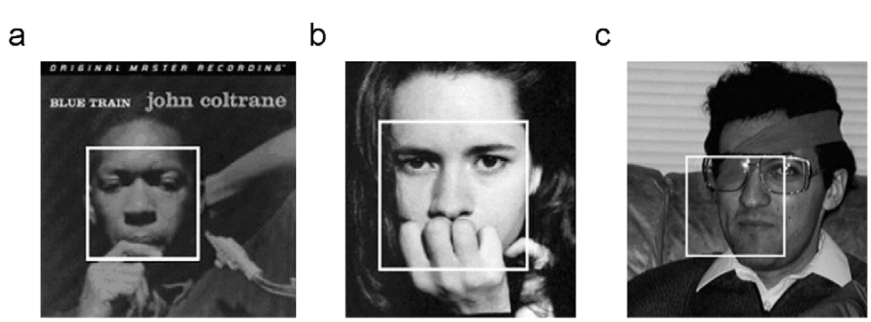



**Fig. 23.** The occluded images.

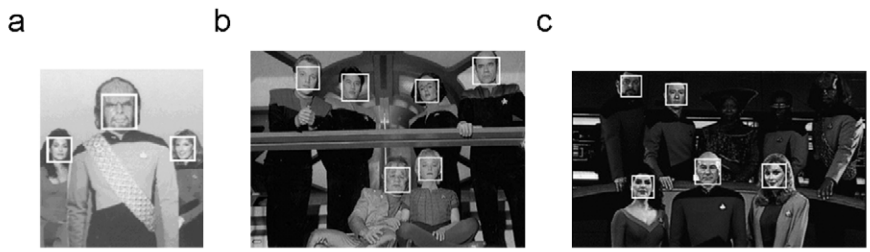



**Fig. 24.** The aliens in the MIT-CMU dataset.


**Fig. 25.** A false positive and a missing face.

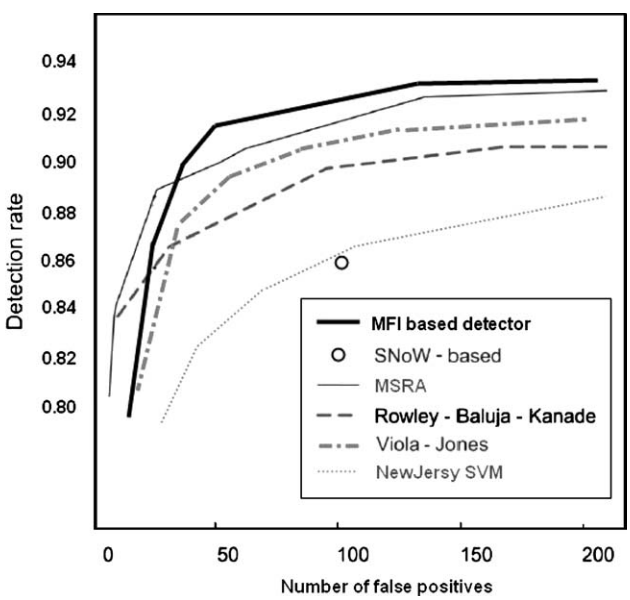



**Fig. 26.** The ROC curve for the CMU-MIT dataset.

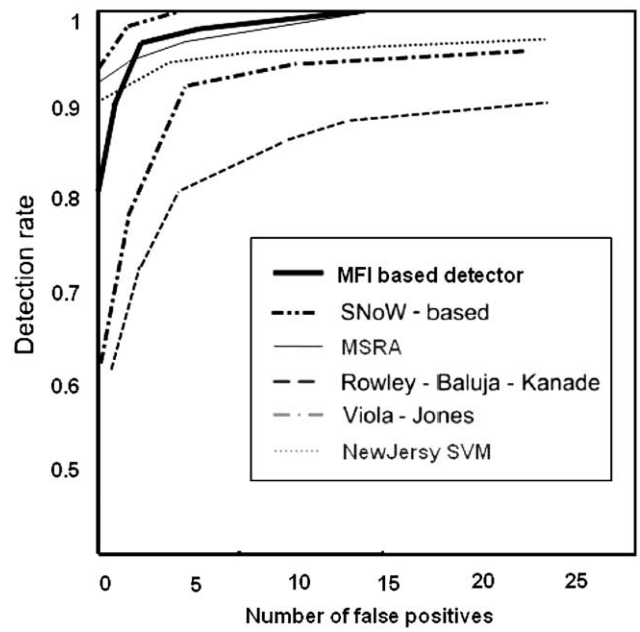



**Fig. 27.** The ROC curve for the BioID dataset.

method can handle more pose variations. We regard our proposed method has a stable and good performance in comparison with the other comparing methods.

We also notice that when the number of false positives is small, the proposed method is not the best but still performs well. This is because some testing images are quite different from the training images used to train the face detector such as the man with moustache, the man wearing glasses, or the faces with dark illumination.

Furthermore, we use the distance based quality measure proposed in [9] to evaluate our method. Let $C_l$ and $C_r$ be the centers of the manually extracted left and right eyes, $C_l'$ and $C_r'$ be the centers of the detected left and right eyes, $d_l$ be the Euclidean distance between $C_l$ and $C_l'$, $d_r$ be the Euclidean distance between $C_r$ and $C_r'$ and $d_{lr}$ be the Euclidean distance between $C_l$ and $C_r$. Then the relative error of detection is defined as $\max(d_l,d_r)/d_{lr}$. A face is considered as correctly detected if the relative error is smaller than a threshold $T$. Fig. 28 shows the detection rate versus the relative error for our proposed method and the comparing methods. It's shown that the proposed method outperforms the comparing methods.

In summary, the experimental results show that the proposed method has outstanding performance in the MIT-CMU dataset and comparable performance in the BioID dataset in terms of both false positive and detection rate. We utilize the integral image approach to compute the variance and sum up the intensity values, apply the feature patterns mined to form the feature vectors, and use the k-d-tree to speed up the SVM. Thus, the proposed algorithm is fast enough for a real-time face detector. Moreover, the proposed method can detect multiple faces in an image, including faces with different scales, slightly rotated faces, and faces in low-quality images.

## 5. Conclusions and future work

We proposed a novel face detection method by integrating the maximal frequent pattern algorithm. An edge image of facial features is derived by applying Sobel's edge detection operator, morphological operator, and thresholding. We use the MAFIA algorithm to mine the maximal frequent patterns from edge and non-edge images of human faces and obtain the positive and negative feature patterns. Based on the positive and negative feature patterns mined, we construct a face detector containing three cascaded classifiers to prune non-face candidates. In the

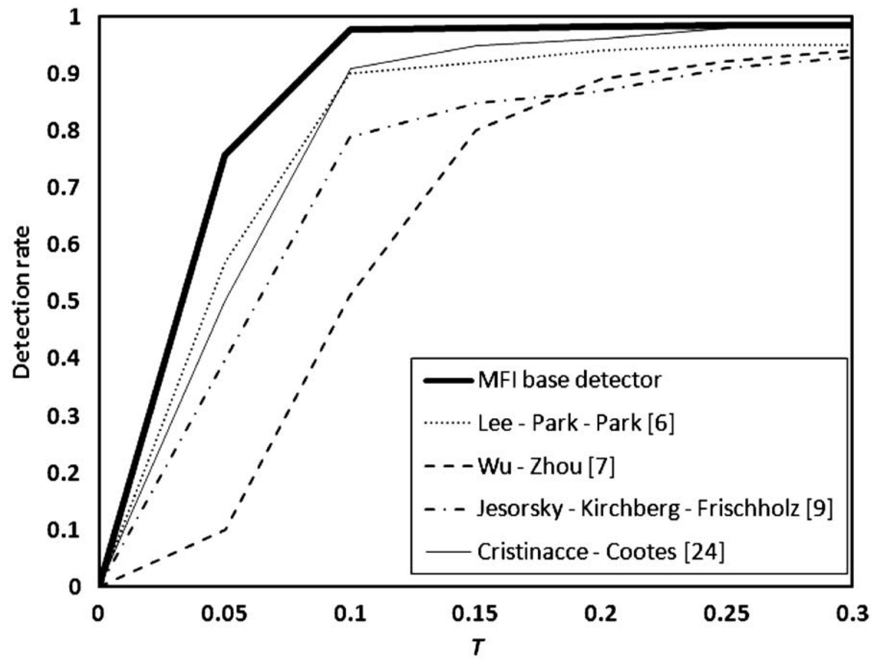



**Fig. 28.** The detection rate vs. the relative error.

detection phase, we apply a window to slide across a testing image in different scales. For each sliding window, we input it to the variance, face feature, and kd-tree-based SVM classifiers. If the sliding window passes these three classifiers, it is considered as a human face.

Since the proposed method not only utilizes the positive feature pattern but also the negative feature pattern, it can capture most of features in a human face. The facial patterns of human faces in terms of edges do not change much in various conditions. Thus, the proposed method is robust to races, illumination, and facial expressions.

The experimental results show that the proposed method has outstanding performance in the MIT-CMU dataset and comparable performance in the BioID dataset in terms of both false positive and detection rate. Besides the better performance, one advantage of the proposed method is that unlike knowledge-based methods, we can obtain the feature patterns without requiring any prior knowledge. We can efficiently obtain the positive and negative feature pattern from the training images using the MAFIA algorithm. Compared to the appearance-based methods, the proposed method does not need a lot of positive and negative training examples and takes very little training time to generate the feature patterns. Furthermore, the feature patterns mined are interpretable. We can relate each part of the feature patterns to the facial components like eyebrows, eyes, nose, mouth, or cheeks. Thus, we can easily modify the feature patterns to fit different applications.

The positive and negative feature patterns are important features to face detection. However, it has some limitations. First, if an image is too blurred or lack of important facial features, we may not be able to extract the features to detect faces. Another limitation is that the feature patterns may change if the poses of faces are largely changed. In the future, we will extend the proposed method to use not only facial components in terms of edges but other information to detect or recognize human faces. Also, we will extend our proposed method to handle rotated faces. Finally, by replacing the training images, we can obtain the maximal frequent patterns of other kinds of objects and then detect them.

## Acknowledgment

The authors are grateful to the anonymous referees for their helpful comments and suggestions. This research was supported in part by the National Science Council of Republic of China under Grant no. NSC 97-2410-H-002-117.

## References

[1] M.H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 34–58.
[2] C.C. Chiang, C.J. Huang, A robust method for detecting arbitrarily tilted human faces in color images, Pattern Recognition Letters 26 (16) (2005) 2518–2536.
[3] C.C. Chiang, W.K. Tai, M.T. Yang, Y.T. Huang, C.J. Huang, A novel method for detecting lips, eyes and faces in real time, Real-Time Imaging 9 (4) (2003) 277–287.
[4] R.L. Hsu, M. Abdel-Mottaleb, A.K. Jain, Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 696–706.
[5] M.A. Bhuiyan, V. Ampornaramveth, S.Y. Muto, H. Ueno, Face detection and facial feature localization for human–machine interface, NII Journal 5 (2003) 25–39.
[6] T. Lee, S.K. Park, M. Park, A new facial features and face detection method for human–robot interaction, in: Proceedings of IEEE International Conference on Robotics and Automation, 2005, pp. 2063–2068.
[7] J. Wu, Z.H. Zhou, Efficient face candidate selector for face detection, Pattern Recognition 36 (11) (2004) 1175–1186.
[8] A. Samal, P.A. Iyengar, Human face detection using silhouettes, International Journal of Pattern Recognition and Artificial Intelligence 9 (6) (1995) 845–867.
[9] O. Jesorsky, K.J. Kirchberg, R.W. Frischholz, Robust face detection using the hausdorff distance, in: Proceedings of International Conference on Audio- and Video-based Biometric Person Authentication, 2001, pp. 90–95.
[10] H. Schneiderman, T. Kanade, A statistical method for 3D object detection applied to faces and cars, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2000, pp. 746–751.
[11] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.
[12] H.A. Rowley, S. Baluja, T. Kanade, Rotation invariant neural network-based face detection, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 1998, pp. 38–44.
[13] L.L. Huang, A. Shimizu, Y. Hagihara, H. Kobatake, Face detection from cluttered images using a polynomial neural network, Neurocomputing 51 (2003) 197–211.

[14] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, in: Proceedings of Computer Vision and Pattern Recognition, 1997.
[15] P. Shih, C. Liu, Face detection using discriminating feature analysis and support vector machine, Pattern Recognition 39 (11) (2006) 260–276.
[16] B. Heisele, Hierarchical classification and feature reduction for fast face detection with support vector machines, Pattern Recognition 36 (9) (2003) 2007–2017.
[17] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, International Journal of Computer Vision 57 (2002) 137–154.
[18] L.L. Huang, A. Shimizu, Y. Hagihara, H. Kobatake, Gradient feature extraction for classification-based face detection, Pattern Recognition 36 (11) (2003) 2502–2511.
[19] L.L. Huang, A. Shimizu, H. Kobatake, A multi-expert approach for robust face detection, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 942–945.
[20] B. Heisele, T. Poggio, M. Pontil, Face Detection in Still Gray Images, A.I. Memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.
[21] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the Second European Conference on Computational Learning Theory, 1995, pp. 119–139.
[22] D. Roth, M. Yang, N. Ahuja, A SNoW-based face detector, in: Advances in Neural Information Processing System 12 (NIPS12), MIT-Press, 2000, pp. 855–861.
[23] M. Nilsson, J. Nordberg, I. Claesson, Face detection using local SMQT features and split up SNoW classifier, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2007, pp. 589–592.
[24] D. Cristinacce, T. Cootes, Feature detection and tracking with constrained local models, in: Proceedings of British Machine Vision Conference, 2006, pp. 929–938.
[25] D. Burdick, M. Calimlim, J. Gehrke, MAFIA: a maximal frequent itemset algorithm for transactional databases, in: Proceedings of International Conference on Data Engineering, 2001, pp. 443–452.
[26] H. Rowley, S. Baluja, T. Kanade, Human face detection in visual scenes, Advances in Neural Information Processing Systems 8 (1996) 875–881.
[27] The Himalaya data mining tools, Cornell Database Group, Cornell University, 〈http://himalaya-tools.sourceforge.net〉.
[28] J.L. Bentley, J.H. Friedman, Data structures for range searching, ACM Computing Surveys 11 (4) (1979) 397–409.
[29] R. Xiao, M.-J. Li, H.-J. Zhang, Robust multipose face detection in images, IEEE Transactions on Circuits and System for Video Technology 14 (1) (2004).
[30] MIT-CMU Face Dataset, 〈http://vasc.ri.cmu.edu/idb/html/face/frontal_images〉.
[31] BioID Face Dataset, 〈http://www.bioid.com/〉.
[32] OpenCV, 〈http://sourceforge.net/projects/opencvlibrary〉.
[33] LibSVM, 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉.
[34] 〈http://vasc.ri.cmu.edu/NNFaceDetector/〉.
[35] 〈http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13701&objectType=FILE〉.

**About the Author**—WEN-KWANG TSAO is currently a Ph.D. candidate in the Department of Information Management, National Taiwan University, Taipei, Taiwan, ROC. He received the B.S. and M.S. degree from Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, in 1997 and 1999, respectively. His current research interests include multimedia databases and data mining.

**About the Author**—ANTHONY J.T. LEE received the B.S. degree from National Taiwan University, Taiwan, in 1983. He got the M.S. and Ph.D. degree in Computer Science from University of Illinois at Urbana-Champaign, U.S.A., in 1990 and 1993, respectively. In August 1993, he joined the Department of Information Management at National Taiwan University and he is now a professor. His current research interests include multimedia databases, temporal and spatial databases, and data mining.

**About the Author**—YING-HO LIU is currently a Ph.D. candidate in the Department of Information Management, National Taiwan University, Taipei, Taiwan, ROC. He received the Bachelor and Master degrees in Business Administration from National Taiwan University, Taipei, Taiwan, in 2001 and 2003, respectively. His research interests include face detection/recognition, data mining, image processing, and multimedia applications.

**About the Author**—TING-WEI CHANG received the Bachelor and Master degrees in Business Administration from National Taiwan University, Taipei, Taiwan, in 2005 and 2007, respectively. His research interests include face detection/recognition, data mining, image processing, and multimedia applications.

**About the Author**—HSIU-HUI LIN is currently a Ph.D. candidate in the Department of Information Management, National Taiwan University, Taipei, Taiwan, ROC. She received the B.S. degree from Department of Computer Science and Information Engineering, National Taiwan University, in 1988, and the M.S. degree from Department of Computer Science and Information Engineering, National Taiwan University, in 1991. Her current research interests include multimedia databases, and digital copyright management.