Pattern Recognition 45 (2012) 3304–3316

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Robust face detection using local gradient patterns and evidence accumulation

Bongjin Jun, Daijin Kim*

Department of CSE, POSTECH, San 31, Hyoja-Dong, Nam-Gu, Pohang 790-784, Republic of Korea

ARTICLE INFO

Article history:
Received 18 August 2011
Received in revised form
18 February 2012
Accepted 23 February 2012
Available online 21 March 2012

Keywords:
Local binary pattern
Local gradient pattern
Face detection
Evidence accumulation

ABSTRACT

This paper proposes a novel face detection method using local gradient patterns (LGP), in which each bit of the LGP is assigned the value one if the neighboring gradient of a given pixel is greater than the average of eight neighboring gradients, and 0 otherwise. LGP representation is insensitive to global intensity variations like the other representations such as local binary patterns (LBP) and modified census transform (MCT), and to local intensity variations along the edge components. We show that LGP has a higher discriminant power than LBP in both the difference between face histogram and non-face histogram and the detection error based on the face/face distance and face/non-face distance. We also reduce the false positive detection error greatly by accumulating evidences from multi-scale detection results with negligible extra computation time. In experiments using the MIT+CMU and FDDB databases, the proposed LGP-based face detection followed by evidence accumulation method provides a face detection rate that is 5–27% better than those of existing methods, and reduces the number of false positives greatly.

1. Introduction

Face detection and recognition is widely used in many applications including biometrics, visual surveillance, human robot interactions, and mobile devices. For such applications, the face must be represented effectively. Many face representation method have been proposed. One of the most successful face representation methods is LBP [1], which has high discriminative power for texture classification due to its invariance to global intensity variations. It uses a 3×3 kernel that summarizes the local structure of an image. At a given pixel position (x_c,y_c), it inspects each of the 3×3 neighborhood pixels surrounding the given pixel and generates a 1 if the neighbor pixel has a value greater than or equal to the given pixel, and a 0 otherwise. The decimal form of the resulting 8-bit word (LBP code) can be expressed as

$$\mathrm{LBP}_{P,R}(x_c,y_c)=\sum_{n=0}^{7} s(i_n-i_c)2^n, \tag{1}$$

where i_c is the pixel value at (x_c,y_c), i_n is one of the eight surrounding pixel values, and the sign function $s(\cdot)$ is defined such that

$$s(x)=\begin{cases}0 & \text{if } x<0,\\ 1 & \text{otherwise,}\end{cases} \tag{2}$$

where the subscripts P and R represent the number of neighboring pixels and the radius in multi-scale LBP, respectively. For example, $\mathrm{LBP}_{8,2}$ denotes the LBP with eight equally spaced pixels on a circle of radius two.

Many variants of LBP have been applied to tasks such as face detection [2–4], face verification [5], face recognition [6–9], facial expression recognition [10,11], human detection [12], gate recognition [13], image retrieval [14], texture recognition [15] and object detection [16].

Census transform (CT) [17] summarizes the local image structure as a bit string, where it is 0 if the intensity value at a position in one image is less than the intensity value at the corresponding position in another image. CT has been extended to the modified census transform (MCT) [18] as

$$\mathrm{MCT}(x_c,y_c)=\sum_{n=0}^{8} s(i_n-\bar{i}_c)2^n, \tag{3}$$

where $\bar{i}_c$ denotes the mean of pixel values in a 3×3 local kernel surrounding (x_c,y_c), and i_n is one of the nine pixel values in the local kernel. The function $s(\cdot)$ is the same as Eq. (2). MCT can be considered to be an enlarged version of the original LBP, which means that one pixel in the image is represented by nine bits. Hence, MCT uses 512 codes, whereas LBP uses 256 codes.

* Corresponding author. Tel.: +82 54 279 2249; fax: +82 54 279 2299.
E-mail addresses: simple21@postech.ac.kr (B. Jun), dkim@postech.ac.kr (D. Kim).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.02.031

Many variants of LBP have been applied to tasks such as face detection [2–4], face recognition [6–8], facial expression recognition [10,11], gate recognition [13], image retrieval [14], texture recognition [15] and object detection [16].

Face representations such as LBP, ULBP, and MCT provide transformed output images that are invariant to the global intensity variations (see Fig. 1).

However, they are sensitive to local intensity variations that occur commonly along edge components such as eyes, eyebrows, noses, mouths, whiskers, beards, or chins due to internal factors (eye glasses, contact lenses, or makeup) and external factors (different backgrounds). This sensitivity of the existing face representation methods generates different patterns of local intensity variations and makes learning of the face detector by AdaBoost difficult. To overcome this problem, we propose a novel face representation method called local gradient patterns (LGP), which generates constant patterns irrespective of local intensity variations along edges. We show that LGP has a greater ability than LBP to determine the difference between face histograms and non-face histograms and to reduce detection error based on the distance between faces and between faces and non-faces.

Furthermore, we propose to use an evidence accumulation method (EAM) technique to reduce the false positive detection error greatly. In EAM, we slide the face detector with the smallest scale over the input image, which provides the confidence value that the detected area is a face at the smallest scale. Then, we slide the face detector with the next larger scale over the input image, which provides the confidence value that the detected area is a face at the next larger scale. We repeat these sliding operations up to the largest scale and accumulate the confidence values at all scales. Accumulating these values makes the confidence that a detected area is a face higher around the real face positions because using several face detectors whose scales are similar to, slightly larger than, or slightly smaller that the real face provides the confidence value that a detected area is a face. A non-face region has a small confidence value of being a face and if it is mistakenly detected as face region is suppressed by the thresholding operation because only one specific size of face detector provides erroneous evidence that the non-face region is a face. We will show that EAM removes many false positive detection errors with negligible computation time.

This paper is organized as follows. Section 2 describes the principle of the proposed LGP and proves that it has higher power to discriminate faces from non-faces than does LBP. Section 3 describes the face detection using LGP and AdaBoost learning. Section 4 describes EAM to reduce the false positive detection error. Section 5 explains the experimental results to demonstrate the detection accuracy of the proposed LGP. Section 6 presents conclusions.

2. Principle of local gradient patterns

The LGP operator uses the gradient values of the eight neighbors of a given pixel, which are computed as the absolute value of intensity difference between the given pixel and its neighboring pixel. Then, the average of the gradient values of the eight neighboring pixels is assigned to the given pixel and is used as the threshold value for LGP encoding as follows. A pixel is assigned a value of 1 if the gradient value of a neighboring pixel is greater than the threshold value, and a value of 0 otherwise. The LGP code for the given pixel is then produced by concatenating the binary 1 s and 0 s into a binary code (see Fig. 2).

The LGP operator is extended to use different sizes of neighborhoods. We consider a circle of radius r centered on a specified pixel and take p sampling points on the circle (see Fig. 3). To obtain the values of pixel positions in the neighborhood for r and p, bilinear interpolation is necessary.

The LGP operator uses a $2\times r+1$ by $2\times r+1$ kernel that summarizes the local structure of an image. At a given center pixel position (x_c,y_c), it takes the $2\times r+1$ by $2\times r+1$ neighboring pixels surrounding of the center pixel. Here, we define the gradient value between the center pixel i_c and its neighboring pixel i_n as $g_n=|i_n-i_c|$, and set the average of p gradient values as $\overline{g}=(1/p)\sum_{n=0}^{p-1} g_n$. Then, $\mathrm{LGP}_{p,r}(x_c,y_c)$ can be expressed as

$$\mathrm{LGP}_{p,r}(x_c,y_c)=\sum_{n=0}^{p-1} s(g_n-\overline{g})2^n, \tag{4}$$

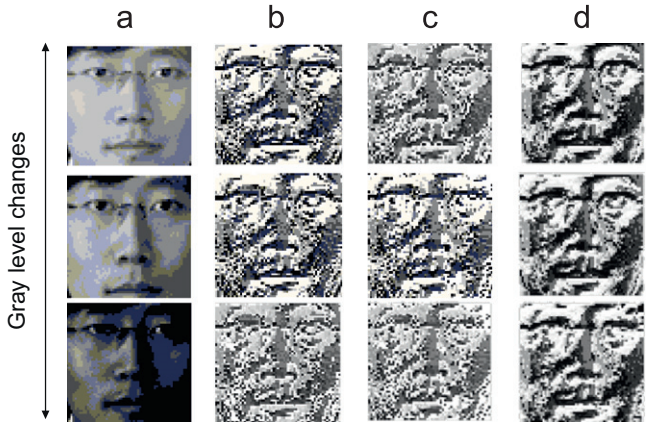

Fig. 1. Robustness to global intensity variations: (a) original image, (b) LBP, (c) ULBP, and (d) MCT.

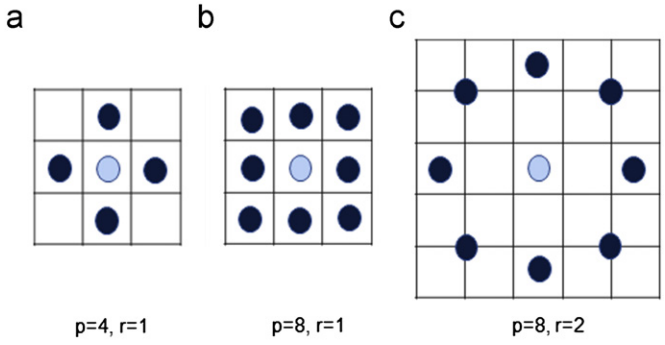

Fig. 3. Three examples of neighboring pixels: $\mathrm{LGP}_{4,1}$, $\mathrm{LGP}_{8,1}$ and $\mathrm{LGP}_{8,2}$. (a) $p=4$, $r=1$, (b) $p=8$, $r=1$, and (c) $p=8$, $r=2$.

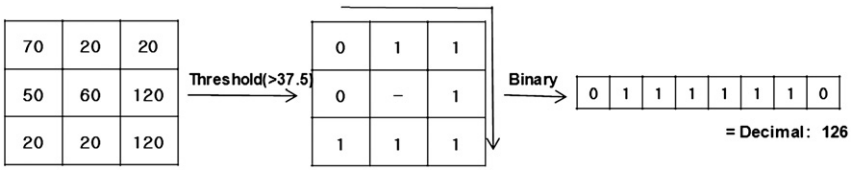

Fig. 2. The original LGP operator.

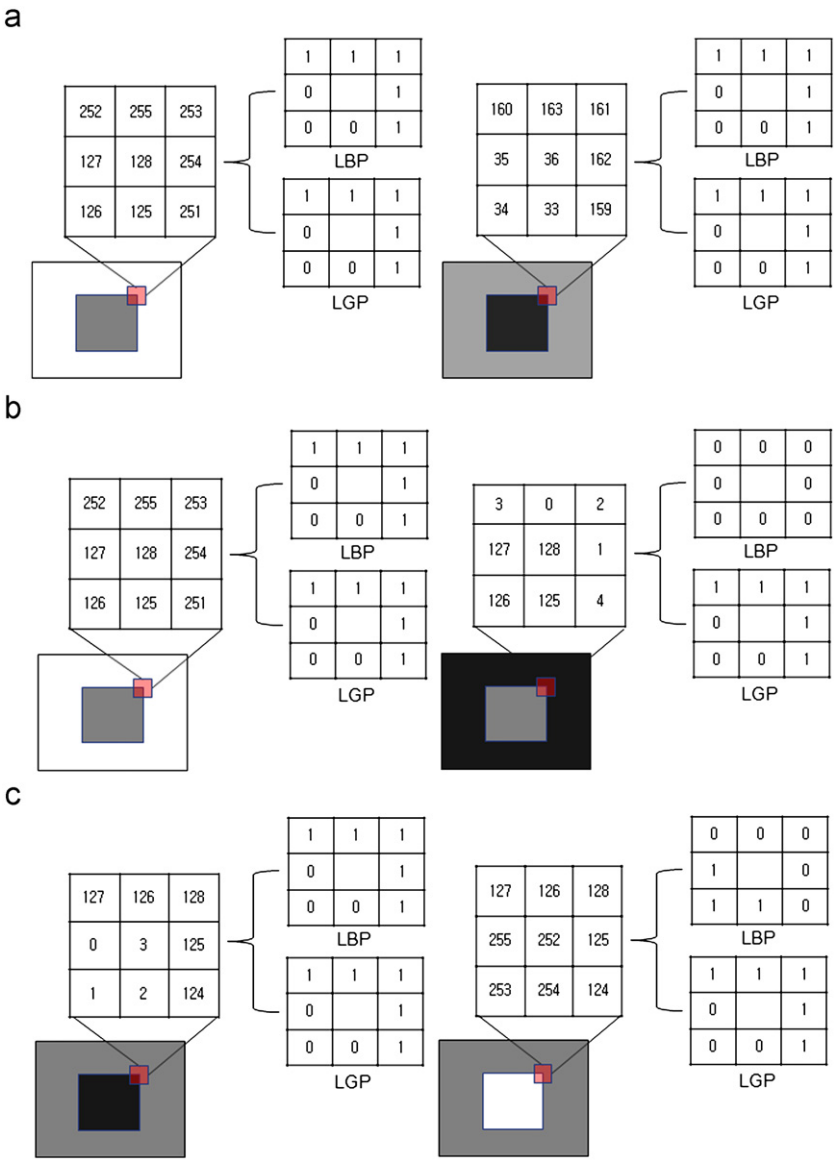

Fig. 4. LBP and LGP patterns when the intensity levels are changed globally or locally. (a) LBP and LGP patterns when monotonic illumination changes, (b) LBP and LGP patterns when the gray level of background changes, and (c) LBP and LGP patterns when the gray level of foreground changes.

where

$$s(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

LGP is invariant to local intensity variations. When the intensity levels of both the background and the foreground are changed together (globally), LGP and LBP both generate invariant patterns (see Fig. 4(a)). However, when the intensity level of the background or the foreground is changed locally, LGP generates invariant patterns but LBP generates variant patterns (see Fig. 4(b) and (c)). This difference occurs because LGP generates patterns using the gradient difference ($s(g_n - \overline{g})$), whereas LBP generates patterns using the intensity difference ($s(i_n - i_c)$).

LGP is less affected by local color variation than LBP. For example, when the local foreground changes, e.g., due to wearing glasses of different colors, LGP always produces the same pattern (00000111) but LBP produces different patterns (00000111 and 00000000). The average number of patterns that have a pattern index of 7 (=00000111) is 1.09 when using LBP and 0.52 when using LGP (see Fig. 5).

LGP has a greater discriminant power than LBP. We show this difference using two metrics: (1) the sum of the differences between face histograms and non-face histograms and (2) detection error based on the face/face distance and face/non-face distance. To compute the discriminant power of two features, we used the face database FDD06,[1] which included 14,818 face images (containing 30,000 faces) and 17,000 non-face images from the internet. We generated 100,000 face images by scaling, rotating, and mirroring randomly and 100,000 non-face images. Each face image was normalized to a size of 22×24 pixels using the center positions of both eyes that were position on (5,6) and (16.6). Each non-face image was obtained by taking image patches with a size of 22×24 pixels randomly.

First, we use the sum of the differences between face histograms and non-face histograms. Given N_f face images and N_{nf} non-face images, we denote the LGP coded face image as G_i^f and

[1] You can find this public face database from http://imlab.postech.ac.kr/faceDB/FDD06/FDD06.html.

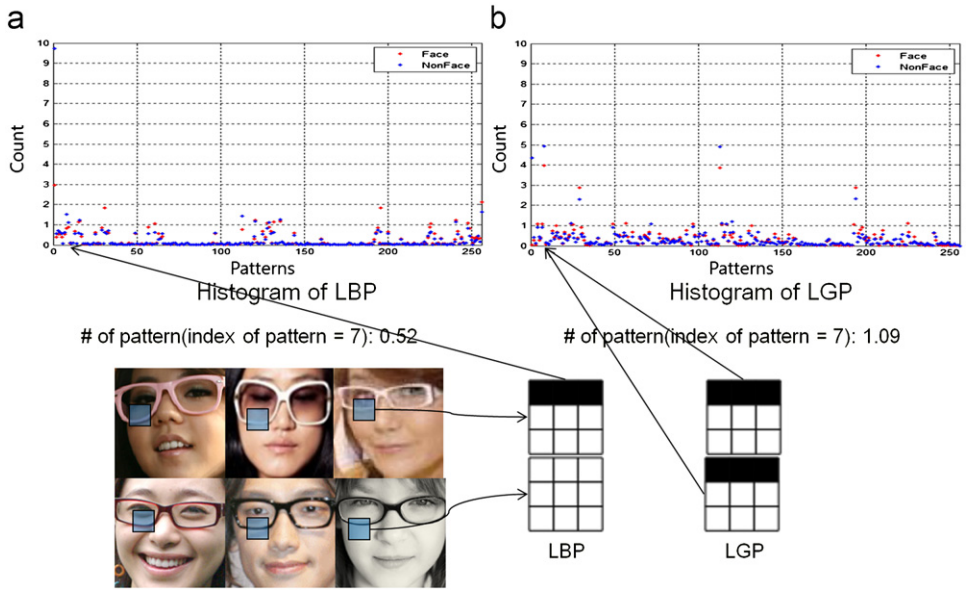

Fig. 5. LGP is less sensitive to local intensity variation than LBP. (a) Histogram of LBP and (b) histogram of LGP.

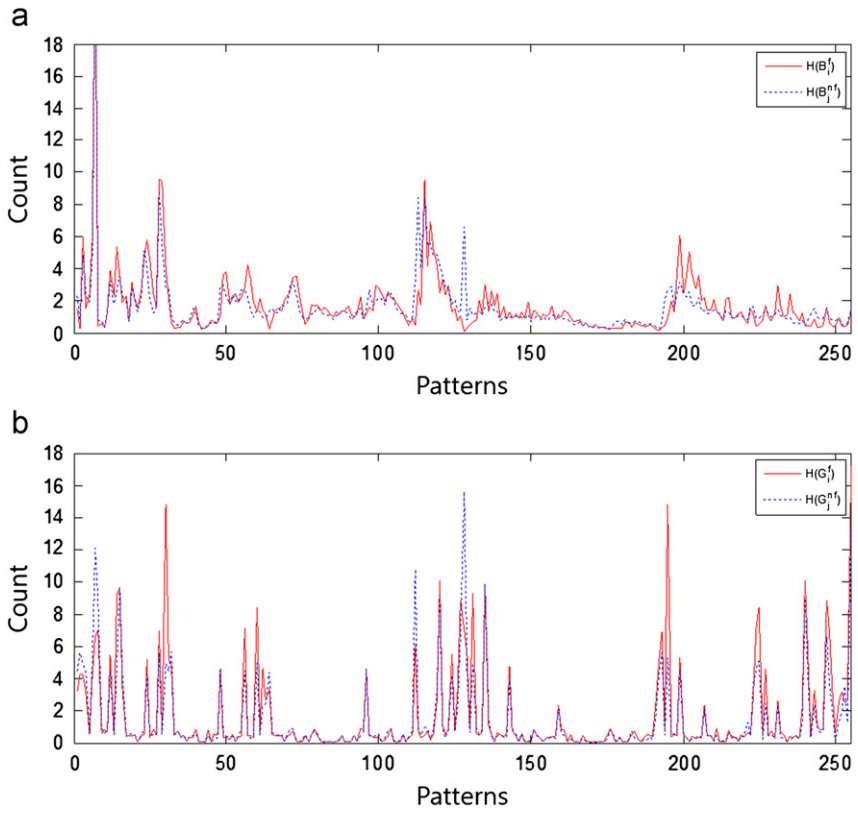

Fig. 6. Average histograms $\overline{H}(B^f)$, $\overline{H}(B^{nf})$, $\overline{H}(G^f)$ and $\overline{H}(G^{nf})$. (a) Average histograms of LBP and (b) average histograms of LGP.

the LGP coded non-face image as G_j^{nf}, the LBP coded face image as B_i^f, the LBP coded non-face image as B_j^{nf}, where $i=1,2,\ldots,N_f$ and $j=1,2,\ldots,N_{nf}$.

We denote the corresponding histograms as $H(G_i^f)$, $H(G_j^{nf})$, $H(B_i^f)$, $H(B_j^{nf})$, respectively, and the corresponding average histograms as

$$\overline{H}(G^f)=\frac{1}{N_f}\sum_{i=1}^{N_f}H(G_i^f) \quad \text{and} \quad \overline{H}(G^{nf})=\frac{1}{N_{nf}}\sum_{j=1}^{N_{nf}}H(B_j^{nf}),$$

$$\overline{H}(B^f)=\frac{1}{N_f}\sum_{i=1}^{N_f}H(B_i^f) \quad \text{and} \quad \overline{H}(B^{nf})=\frac{1}{N_{nf}}\sum_{j=1}^{N_{nf}}H(B_j^{nf}). \tag{6}$$

Then, we define the powers of LGP and LBP to discriminate between the face images and non-face images as

$$P_{LGP}=\sum_{k=0}^{255}(\overline{H}_k(G^f)-\overline{H}_k(G^{nf}))^2, \quad P_{LBP}=\sum_{k=0}^{255}(\overline{H}_k(B^f)-\overline{H}_k(B^{nf}))^2, \tag{7}$$

where k is the histogram bin index.

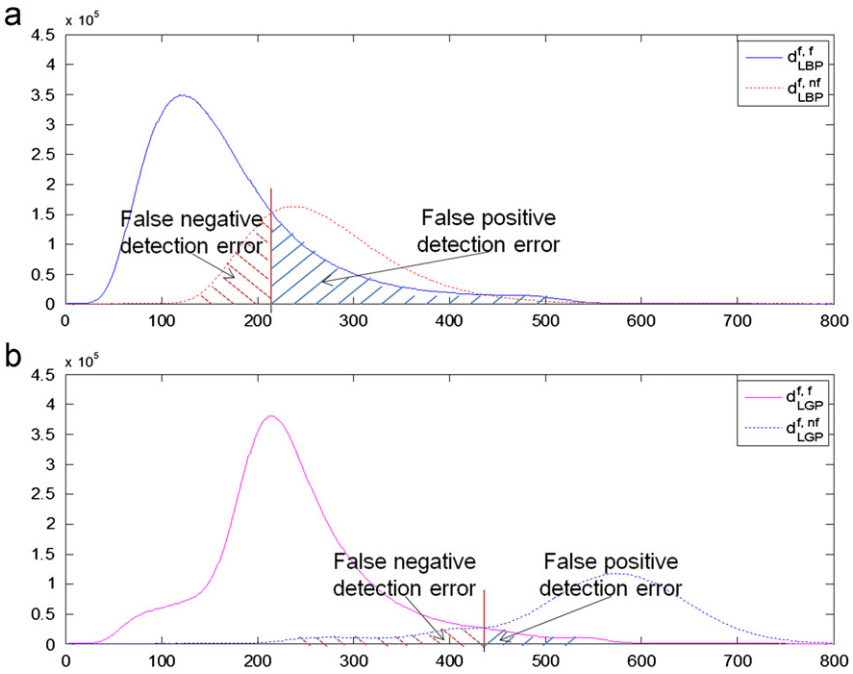

Fig. 7. LGP has smaller detection error than LBP. (a) The face/face and the face/non-face distance distribution of LBP and (b) the face/face and the face/non-face distance distribution of LGP.

We obtained average histograms $\overline{H}(G^f)$, $\overline{H}(G^{nf})$, $\overline{H}(B^f)$ and $\overline{H}(B^{nf})$ from 100,000 face images and 100,000 non-face images (see Fig. 6). LGP showed peaks at more patterns than LBP; this means that LGP produced patterns that are more coherent over the local intensity variation than LBP. In this case, LGP has a discriminant power $P_{LGP}=2695$ and LBP has a discriminant power $P_{LBP}=516$.

Second, we use the detection error based on the face/face distance and face/non-face distance. We define the face/face distance of LGP and LBP between the ith face image and the jth face image as

$$d^{f,f}_{G_i,G_j}=\sum_{k=0}^{255}(H_k(G_i^f)-H_k(G_j^f))^2,\quad d^{f,f}_{B_i,B_j}=\sum_{k=0}^{255}(H_k(B_i^f)-H_k(B_j^f))^2. \tag{8}$$

Similarly, we define the face/non-face distance of LGP and LBP between the ith face image and the jth non-face image as

$$d^{f,nf}_{G_i,G_j}=\sum_{k=0}^{255}(H_k(G_i^f)-H_k(G_j^{nf}))^2,\quad d^{f,nf}_{B_i,B_j}=\sum_{k=0}^{255}(H_k(B_i^f)-H_k(B_j^{nf}))^2. \tag{9}$$

Then, we compute the face/face distances of LGP and LBP histograms for all possible pairwise combinations face images and compute the face/non-face distances of LGP and LBP histograms for all possible combinations of one face image and another non-face image. Then, we plot the face/face and face/non-face distance distributions (see Fig. 7(a) and (b)). LGP has longer face/face distance and longer face/non-face distance than LBP, and LGP has smaller detection error than LBP because the left and right overlapping regions correspond to the false negative detection error (face identified as non-face) and the false positive detection error (non-face identified as face).

3. Face detection using local gradient patterns

Many methods of face detection have been developed. Most approaches are based on the principal component analysis [19], support vector machines [20,21], neural networks [22], maximum likelihood [23], hidden Markov models [24], or Gaussian mixture models [25]. However, those approaches require a large amount of computation time and are sensitive to illumination changes because they use the input color or gray-level images directly.

Recently, a novel face detection algorithm using Harr-like features combined with AdaBoost learning was proposed [26,27]. This method has low sensitivity to global illumination features and achieves remarkably good accuracy and speed. It has been widely used for practical and real-time applications in many fields such as digital media (cell phone, smart phone, smart TV, digital camera), intelligent user interfaces (Wii, MS Kinnect), intelligent visual surveillance, and interactive games.

However, face detection is still a difficult task due to the high variability of face appearances caused by internal factors such as different shapes, colors, textures and facial expressions, and external factors such as different head poses, locally changing illuminations (contrast, shadow), occlusions (glasses) and other facial features (make-up, beard). To make face detection less sensitive to these variations, we propose a novel face detection method using LGP feature, AdaBoost learning and EAM. The proposed face detection method consists of four stages as follows (see Fig. 8).

(1) We transform the input into different scaled images using the pyramid image generation technique [22] to cope with the different face sizes. In real implementations, the original image is repeatedly reduced by a factor of 0.89 ($=\frac{1}{1.125}$). (2) We transform each scaled image into the corresponding LGP feature image to reduce sensitivity to the local and global intensity variations. (3) We scan each LGP feature image with a fixed size of detection window (20×22) by one pixel along the row and then one pixel along the column (raster scan). We apply a cascade of face classifiers [18,26,28] to each detection window to determine whether the detection window is face or not; each stage classifier is trained using the AdaBoost learning method. (4) We make a final decision about whether the image is a face by using EAM, which uses the detection results from several scaled LGP feature images, which will be explained in the next section.

Fig. 8. Overall procedure of the proposed face detection.

The proposed face detection method uses a cascade of classifiers that can improve the face detection accuracy while reducing the face detection time effectively. The key idea of the cascade of classifiers is that the simple and efficient classifier is executed at an early stage to reject most non-faces while detecting most faces, and that more-complicated classifiers follow the previous simple classifier to reduce the number of false positive errors. This early rejection of non-faces reduces the detection time greatly because most detection windows in the input image are non-faces. In this work, we use a four-stage cascade of classifiers (Fig. 8), in which each classifier is designed to obtain a 99% detection rate and a 4% false positive error rate. Therefore, the total detection rate of the cascade is about $(0.99)^4 \approx 96\%$ and the total false positive error rate of the cascade is about $(0.04)^4 \approx 2.56 \times 10^{-6}\%$.

Let $H_j(\mathbf{G})$ be the strong classifier of the jth stage, where $\mathbf{G}$ is an LGP feature vector whose size is 20×22 of one detection window. Then, it is represented by the sum of weak classifiers as

$$H_j(\mathbf{G}) = \sum_{\mathbf{x} \in S} h_{\mathbf{x}}(\mathbf{G}(\mathbf{x})), \tag{10}$$

where S is the set of selected feature points, $\mathbf{x}$ is the selected feature point $\mathbf{x}=(x, y)$, and $h_{\mathbf{x}}(\bullet)$ is the weak classifier at the selected feature point $\mathbf{x}$ that consists of a lookup table with a dimensionality of 256 {0,255} whose index is just the LGP value. In the lookup table, each value of each index is a confidence weight such that a window is likely to be non-face as the weight value becomes increases, and more likely to be a face as the value decreases. The weak classifiers are constructed using AdaBoost learning [29,30], which updates the weight of each training sample such that misclassified instances are given a higher weight in the subsequent iteration. A detailed explanation of the AdaBoost learning procedure of weak classifiers is as follows:

1. Prepare N_f training face images and N_{nf} training non-face images.
2. Apply LGP to all face and non-face images. Let G_m^f and G_n^{nf} be the training face and non-face LGP feature images, respectively, where $m = 1, \ldots, N_f$ and $n = 1, \ldots, N_{nf}$.
3. Initialize the weights of the training face and non-face LGP feature images as $w_1^f(m) = 1/2N_f$ and $w_1^{nf}(n) = 1/2N_{nf}$, define the set of selected feature points $S_1 = \{\}$, and set the values of the weak classifier $h_{\mathbf{x}}(\gamma) = 0$, where $\mathbf{x}$ denotes one of 20×22 LGP feature points and $\gamma = 0, \ldots, 255$.
4. For $t = 1, \ldots, T$
   (a) Generate the weight tables from the training face and non-face LGP feature images as

$$W_t^f(\mathbf{x},\gamma) = \sum_{m=1}^{N_f} w_t^f(m) I(G_m^f(\mathbf{x}) = \gamma),$$

$$W_t^{nf}(\mathbf{x},\gamma) = \sum_{n=1}^{N_{nf}} w_t^{nf}(n) I(G_n^{nf}(\mathbf{x} = \gamma),$$

   where $I(\bullet)$ is an indicator function that takes a value of 1 if the argument is true, and 0 otherwise.
   (b) Compute the error $\epsilon_t(\mathbf{x})$ for each lookup table as

$$\epsilon_t(\mathbf{x}) = \sum_{\gamma} \min\{W_t^f(\mathbf{x},\gamma), W_t^{nf}(\mathbf{x},\gamma)\}.$$

   (c) Select the best feature point $\mathbf{x}_t$ as

$$\mathbf{x}_t = \begin{cases} \mathbf{x} = \min_{\mathbf{x}} \epsilon_t(\mathbf{x}) & \text{if } |S_t| < N_p, \\ \mathbf{x} = \min_{\mathbf{x} \in S_t} \epsilon_t(\mathbf{x}) & \text{otherwise,} \end{cases}$$

   where N_p is the allowed number of selected feature points and S_t is the set of selected feature points until iteration t, thus $S_{t+1} = \{S_t \cup \mathbf{x}_t\}$.
   (d) Select the type of weak classifier at the selected feature point $\mathbf{x}_t$ according to the sum of weights as

$$z_t(\gamma) = \begin{cases} 0 & \text{if } W_t^f(\mathbf{x}_t,\gamma) > W_t^{nf}(\mathbf{x}_t,\gamma), \\ 1 & \text{otherwise.} \end{cases}$$

   (e) Update the weak classifier at the selected feature point $\mathbf{x}_t$ as

$$h_{\mathbf{x}_t}(\gamma) = h_{\mathbf{x}_t}(\gamma) + \alpha_t z_t(\gamma),$$

   where $\gamma = 0, \ldots, 255$ and $\alpha_t = (1/2)\ln((1-\epsilon_t)/\epsilon_t)$.
   (f) Update the weights of the training face and non-face LGP feature images as

$$w_{t+1}^f(m) = w_t^f(m) \cdot \begin{cases} e^{-\alpha_t} & \text{if } z_t(G_m^f(\mathbf{x}_t)) = 0, \\ e^{\alpha_t} & \text{otherwise,} \end{cases}$$

$$w_{t+1}^{nf}(n) = w_t^{nf}(n) \cdot \begin{cases} e^{-\alpha_t} & \text{if } z_t(G_n^{nf}(\mathbf{x}_t)) = 1, \\ e^{\alpha_t} & \text{otherwise,} \end{cases}$$

$$w_{t+1}^{f}(m) = \frac{w_{t+1}^{f}(m)}{D_{t+1}} \quad \text{and} \quad w_{t+1}^{nf}(n) = \frac{w_{t+1}^{nf}(n)}{D_{t+1}}, \tag{11}$$

where $D_{t+1} = \sum_{m=1}^{N_f} w_{t+1}^{f}(m) + \sum_{n=1}^{N_{nf}} w_{t+1}^{nf}(n)$.

5. The final strong classifier is the sum of weak classifiers as

$$H(\mathbf{G}) = \sum_{\mathbf{x} \in S_T} h_{\mathbf{x}}(\mathbf{G}(\mathbf{x})), \tag{12}$$

where S_T is the set of selected feature points at the final iteration T.

4. Improving detection performance by evidence accumulation

Most existing face detection methods try to find the optimal model parameters of the learning algorithms or to devise a novel detection algorithm to reduce the detection error such as false positive and false negative detection error. However, because the model parameters are determined by the face and non-face images in the training set, the algorithms may not work well for novel images. In addition, training every non-face image in natural scene is almost impossible because the real world includes a huge number of non-face images. As a result, face detection algorithms that show good detection accuracy during the training phase may show high false positive and false negative error rates in real environments.

Because we do not know the scale and position of the input faces in advance, we apply a fixed sized face detector to the multiple scaled input images that are generated by the pyramid image construction and scan the face detector along the row and the column for a given scaled input image. Then, the face detector determines whether the detection window in the current scaled input image is face or non-face by testing whether or not the confidence value is above the threshold. Fig. 9 shows the face detection results that are obtained using the proposed method. It shows that (1) the false negative detection error rate is very low due to the high accuracy of the cascade of several face detectors, (2) the false positive detection error rate is relatively low due to the high detection rate of the proposed face detector, (3) two typical situations occur in which several detection windows are located around a real face, but are of slightly different sizes and are slightly displaced from each other (see Fig. 10), and (4) few detection windows occur around falsely accepted faces.

From these observations, we propose a simple way of reducing the false positive detection error (see Table 1); this method accumulates evidence from the detection results of multiple scaled pyramid images and determines that regions in which many detection windows overlap are real face regions, and that regions in which few detection windows overlap are falsely accepted face regions. This method does not require additional

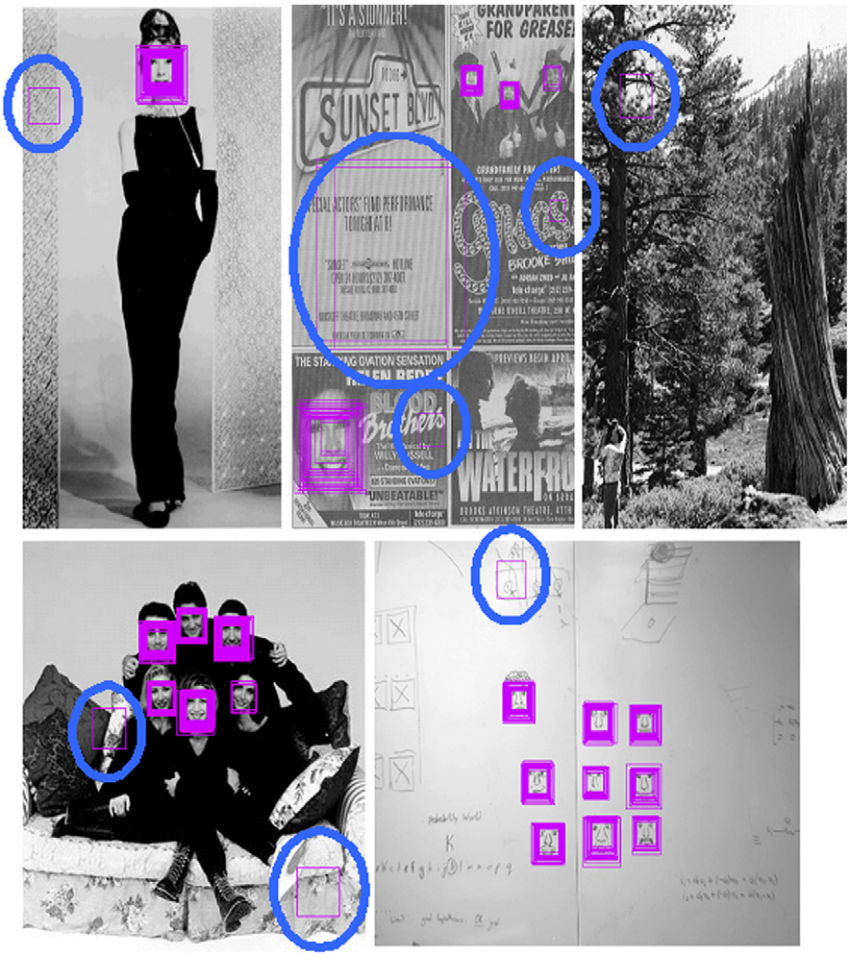

Fig. 9. Some face detection results.

processing time because the proposed face detection method has already scanning multiple differently scaled images and just accumulates the detection result of each scaled image. The similar approaches merge overlapping detections at different locations and scales and using the number of the overlapping detections as a heuristic to prune false detections is a general method that has to be applied in any scanning-window face detector. Rowley [22] and Viola [26] counted the number of detections within a specified neighborhood and the region whose sum is above a threshold is regarded as a face. However, these approaches did not consider the confidences of detector in the each overlapped regions. As an alternative way, Rodriguez [31] and Garcia [32] considered the confidence of detectors to reduce the false positive detection error and the region whose confidence sum is above a threshold is regarded as a face. However, these approaches only considered the confidences with respect to the detected regions. The proposed face detector with EAM considers the confidences over all image regions because the non-detected regions near the real face have the high confidence value.

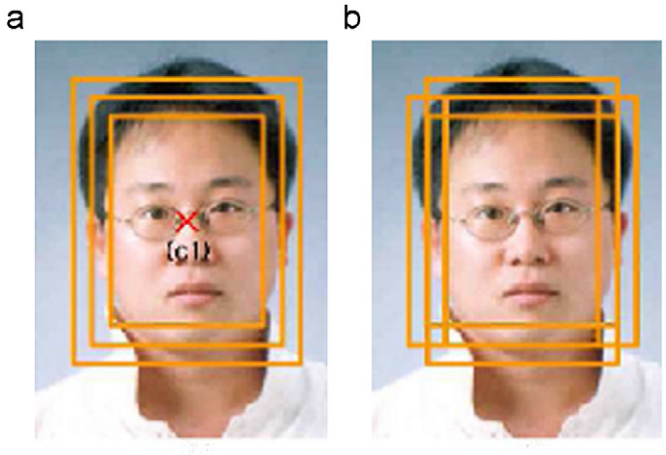

Fig. 10. Two typical face detection results: (a) Detection windows have the same center position (C1) but different scales and (b) detection windows have the same scale but different center positions.

Table 1
Evidence accumulation method. N_x, number of pixels along the x-axis; N_y, number of pixels along the y-axis; N_s, number of scales; N_c, number of cascades; N_p, number of feature points per stage; h_p is a weak classifier at the pth selected feature point that consists of a 256 size of lookup table; $\mathbf{G}(x_p,y_p)$ represents the LGP feature at the pth selected feature point.

Step 1	Initialization: $C(x,y)=0$
Step 2	For $x=1,2,\ldots,N_x$ for x positions
Step 3	For $y=1,2,\ldots,N_y$ for y positions
Step 4	$C_{ij}(x,y)=\sum_{p=1}^{N_p} h_p(\mathbf{G}(x_p,y_p))$ for feature points
Step 5	$C_i(x,y)=\sum_{j=1}^{N_c} C_{i,j}$ for cascades
Step 6	$C(x,y)=\sum_{i=1}^{N_s} C_i(x,y)$ for scales
Step 7	If $C(x,y)$ is a local maximum and is greater than T, then we take the detection window as the face region

Step 1 initializes the confidence value. Steps 2 and 3 iterate the computation of confidence value along the x- and y-axis, respectively. Step 4 computes the confidence value of each stage at each pixel position. Step 5 accumulates the confidence values of all stages. Step 6 accumulates the confidence values of all pyramid images, and the pixel positions (x,y) of each down-scaled pyramid image are aligned to their corresponding original image locations. Step 7 determines the face region by finding the peak whose accumulated confidence value is greater than the threshold value T, then taking the position and the size of detected face window as the peak (maximal) position (x,y) and the detection window at the peak position, respectively. This process suppresses the detection windows of falsely accepted non-faces because their confidence values are usually not high. After finding one face region, we set the confidence value within the detection window to 0 and repeat Step 7 for finding the next face region.

The falsely accepted face windows are suppressed by performing the threshold operation on the confidence value that is obtained by accumulating the evidence from many detection results from the pyramid images; this process results in great reduction in the number of false positive errors (see Fig. 11).

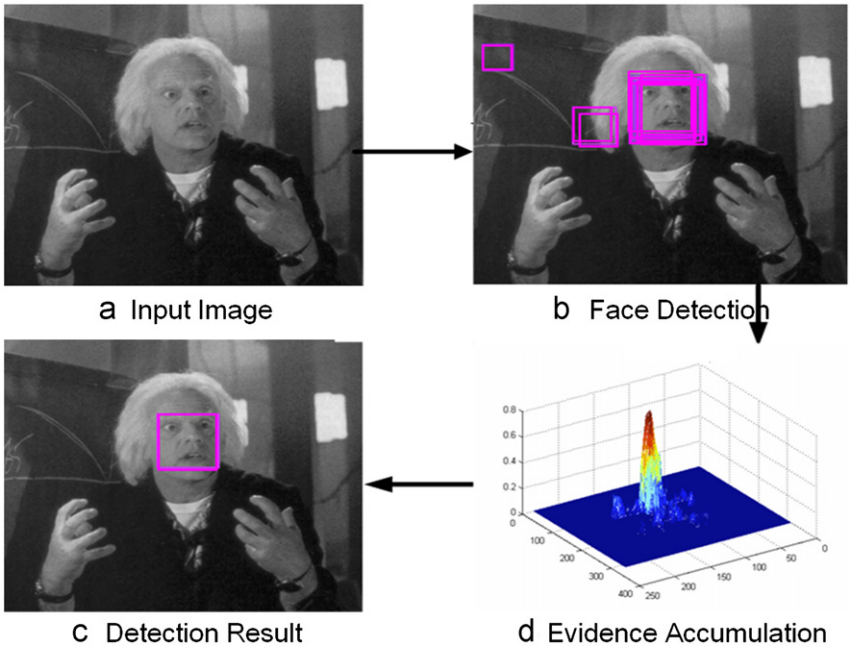

Fig. 11. An illustration of suppressing the detection windows of falsely accepted non-faces.

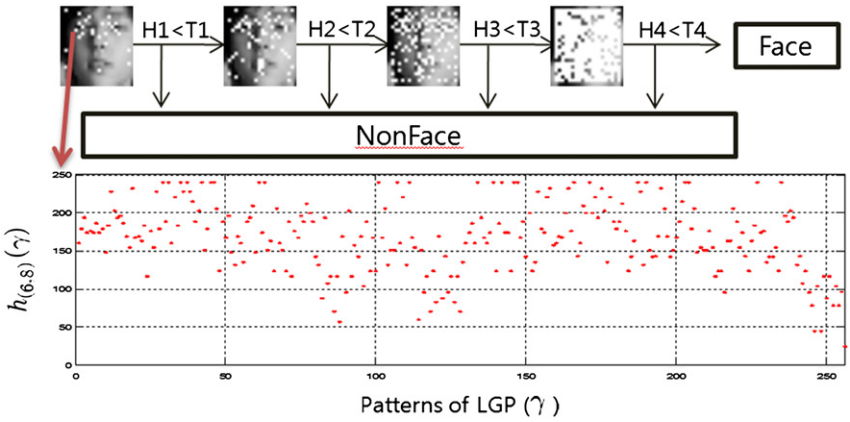

Fig. 12. Four-stage classifiers together with the visualization of a pixel-classifier lookup table: $h_{\mathbf{x}}(\gamma)$ is a confidence weight value that represents a degree of face and non-face; white dots denote positions of weak classifiers for each stage.

5. Experimental results and discussion

5.1. Training of face detector

We collected 30,000 training face images from the internet and annotated them,[2] which contained multiple human types and varied in numerous ways, including illumination conditions, color and texture. Each image was normalized to a size of 22×24 pixels using the center positions of both eyes that were position on (5,6) and (16.6). We gathered another 300,000 training face images that were generated by slightly shifting, scaling with 0.95, 1.0, and 1.05 scale-factors, and rotating the original face images by -15°, 0°, and 15° to test the method's ability to detect faces which did not exactly fit the scanning window. In addition, we doubled the number of training face images by mirroring each face image.

We collected 17,000 images from the internet that did not include faces, and extracted image patches from these images by taking samples of random sizes and at random positions. Then, we generated 300,000 training non-face images by resizing the extracted image patches to a size of 22×24 pixels and used these 300,000 non-face images as the training non-face data for the first stage of the cascaded face detector. For the next stages of the cascaded face detector, we used false positive samples for the training non-face data, where they were the non-face images that were determined to be a face during the previous stage of the cascaded face detector.

We also prepared a validation set of 150,000 face images and 250,000 non-face images in the same way that we used for collecting the training face and non-face images. The validation data set consisted of face and non-face images that were totally different from the training data and was used to determine the threshold value and the stop condition of each stage of the cascaded face detector.

The proposed four-stage cascaded face detector was trained as follows (see Fig. 12). First, we transformed each face and non-face image into its corresponding face and non-face LGP feature image. Although the original face and the non-face images were 22×24, the face and non-face LGP feature images were 20×22 because we performed the LGP operation in the neighborhood of 3×3 window and thus excluded the outer area of each face and non-face image. Then, we selected the feature points for face/non-face classification and the weak classifiers at the selected feature points and determined the threshold value of each stage such that each stage achieved 99% detection rate and 4% false positive error rate when using the validation data set.

Table 2
Number of feature points, threshold, and AdaBoost learning time of each stage.

Cascade	Number of feature points	Threshold	AdaBoost l earning time
1 Stage	26	2179	≈1 min
2 Stage	60	2739	≈5 min
3 Stage	120	3419	≈30 min
4 Stage	400	5699	≈23 h

We set the maximum number of weak classifiers for stages 1 to 4 as 26, 60, 120, and 400, respectively, to achieve 99% detection rate and 4% false positive error rate. Training the proposed four-stage cascaded face detector takes about one day on a 2.83 GHz Intel Pentium IV PC system with 8 GB RAM; stages 1 to 4 take about 1 min, 5 min, 30 min and 23 h, respectively (see Table 2).

5.2. Detection performance

After training the proposed four-stage cascaded face detector, we evaluated the detection accuracy using two kinds of face databases: the MIT+CMU database [33,23] (130 images with 483 faces containing the test sets A, B, C (test, test-low, new-test) without the rotated test set), the Face Detection Data Set and Benchmark (FDDB[3]) database [34] (2845 images with 5171 faces). The face images in the MIT+CMU database are easy to detect because they are frontal and upright and because the illumination change is mild. The face images in the FDDB database are very difficult to detect because occlusions and variation in pose and illumination.

Receiver operating characteristic (ROC) curves were obtained using different approaches, i.e., Rowley–Baluja–Kanade [22], Viola–Jones [26], MB-LBP [6], LBP [1], Mikolajaczyk et al. [35], Subburaman et al. [36], LGP and LGP+EAM. All methods were applied to both the MIT+CMU and FDDB databases.

When using the MIT+CMU database, we obtain that (1) the ROC curve of the proposed LGP+EAM method was the highest among all face detection methods, which means that LGP+EAM shows the best face detection, (2) the number of false positives of the LGP+EAM, LGP, LBP, MB-LBP, Viola–Jones, and Rowley–Baluja–Kanade at the 90% detection rate were 12, 26, 102, 57, 78, and 169, respectively, which means that LGP+EAM produced the

[2] We call this face database FDD06 database (http://imlab.postech.ac.kr/faceDB/FDD06/FDD06.html), which is publicly opened.

[3] See http://vis-www.cs.umass.edu/fddb/results.html.

smallest number of false positives, (3) the proposed LGP+EAM method produced 14 fewer false positives than the LGP method, which is a 10.77% reduction in false positives per image (FPPI), and (4) the smallest number of false positives of the proposed LGP+EAM was 3 when the detection rate was 88% (see Fig. 13(a)).

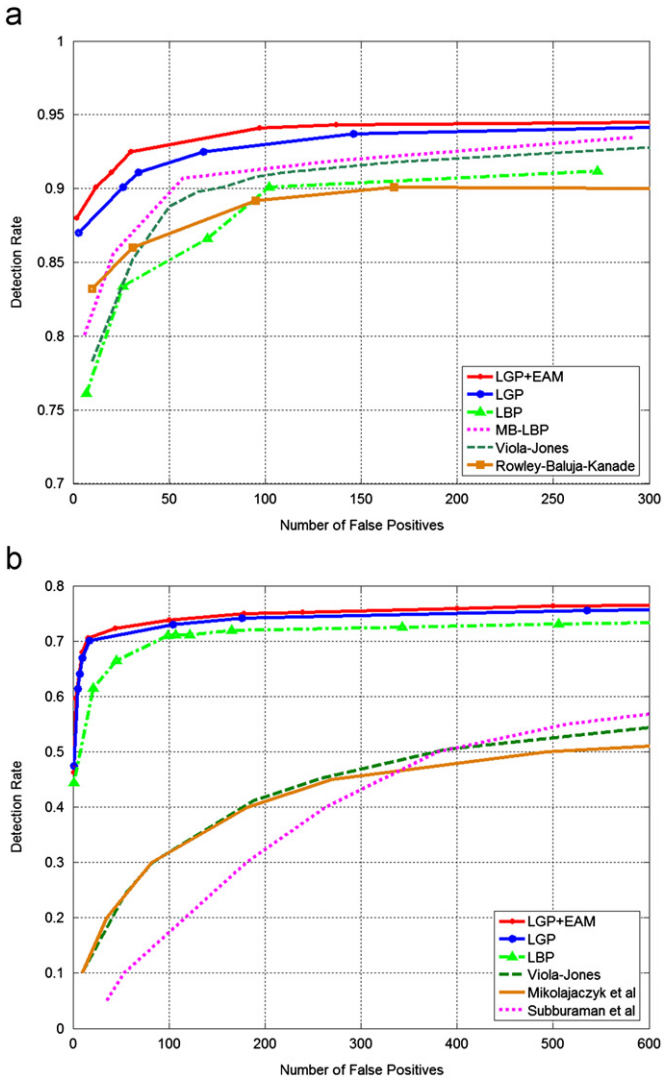

Fig. 13. ROC curves using (a) MIT+CMU database and (b) FDDB database.

When using the FDDB database, we obtain that (1) the ROC curves were lower than those obtained using the MIT+CMU database because the face images in the FDDB database are more difficult to detect than those in the MIT+CMU database, (2) the ROC curve of the proposed LGP+EAM method was also the highest among all face detection methods, which means that LGP+EAM method shows the best face detection, (3) the detection rate of the proposed LGP+EAM at 600 false positives was 76%, which is 20% higher than that of the best existing method by Subburaman et al., (4) the number of false positives of the LGP+EAM, LGP, LBP, Viola–Jones, Mikolajaczyk et al., and Subburaman et al. methods at the 75% detection rate were 123, 220, 868, 16,555, 509,242, and 65,345, respectively, which means that LGP+EAM produced the smallest number of false positives, (5) LGP+EAM produced 97 fewer false positives than did the LGP method, which means that EAM reduced the number of false positives effectively, and (6) the smallest number of false positives of the proposed LGP+EAM was 1 when the detection rate was 32% (see Fig. 13(b)).

From these results, we conclude that the proposed LGP+EAM method shows the best detection accuracy and the smallest number of false positives with negligible extra computation time.

Fig. 14(a) and (b) shows some face detection results using LGP+EAM on the MIT+CMU database and the FDDB database, respectively. From Fig. 14, we observe that (1) LGP+EAM shows a high rate of face detection under a range of size variations, pose variations, illumination changes, racial variations, and occlusions, (2) it shows a small number of false positives, and (3) it can detect a tiny face of 20×20 pixels.

Fig. 15(a) and (b) compares the face detection results using LBP-based and LGP-based face detection methods, respectively. It reveals that in local areas which include a variety of intensity changes, the LBP-based face detection method can fail to detect faces because the LBP values are inconsistent, but that the LGP-based face detection method can detect those faces because the LGP values are relatively consistent. Four typical cases in which the LBP-based face detection method fails to detect faces but the LGP-based face detection method succeeds (Fig. 15) are (1) a locally changing illumination (top left), (2) local intensity changes due to makeup (top right), (3) local intensity change due to wearing white eye glasses (bottom left) and (4) a variety of backgrounds (bottom right).

5.3. Memory size and computation time

Each weak classifier must store the confidence value at each LGP value in the lookup table, where the confidence value is represented by a real number, which consists of 8 bytes. Therefore, each weak

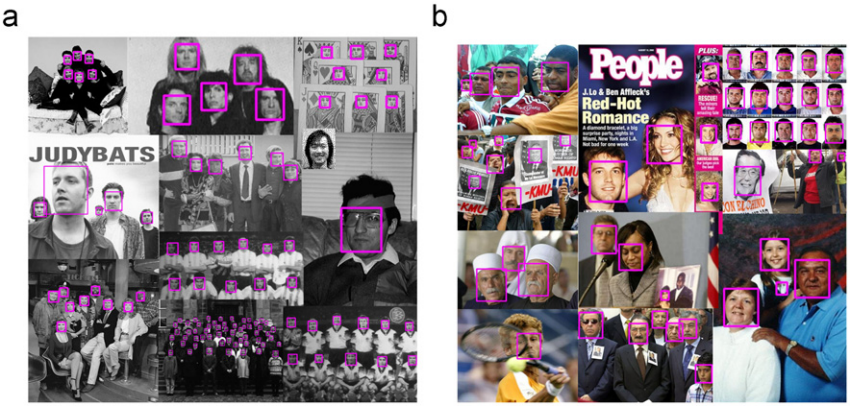

Fig. 14. Some face detection results using (a) MIT+CMU database and (b) FDDB database.

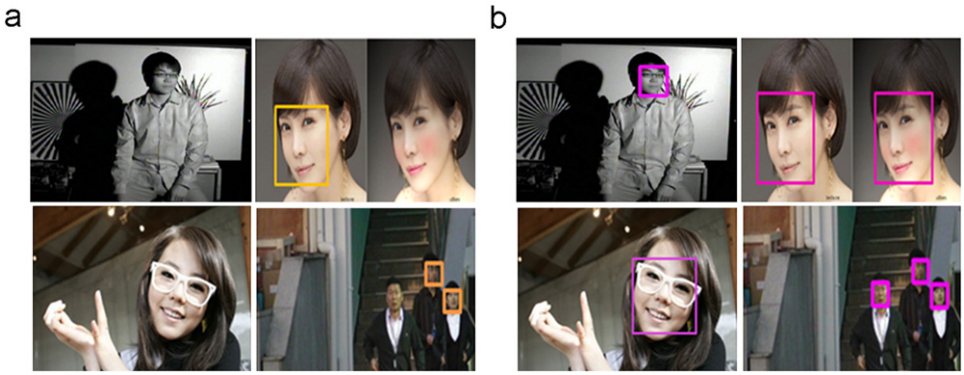

Fig. 15. Comparison of face detection results using (a) LBP-based and (b) LGP-based face detection method.

Table 3
Comparison of average computation time among several face detectors (unit: 10^{-3} s).

Detector	Pyramid image	Transformation	Detection	EAM	Average computation time
Viola–Jones	0.0	0.16	70.06	–	70.22
MCT-based	1.7	1.92 (MCT)	6.15	–	9.77
LBP-based	1.7	1.76 (LBP)	6.20	–	9.66
LGP-based	1.7	2.05 (LGP)	6.07	0.3	10.12
Rowley–Baluja–Kanade	–	–	–	–	1053.3
Schneiderman–Kanade	–	–	–	–	42,132.0

classifier requires a memory space of 2048 bytes (=256 LGP patterns $\times$ 8 bytes). Because stages 1 to 4 consist of 26, 60, 120 and 400 weak classifiers respectively, the total required memory space is 1.2 MB (=606 $\times$ 2048 bytes), which is a burden for low-performance embedded systems. Furthermore, most low-performance embedded systems do not support the floating point operation. To overcome this limitation, we propose an encoding scheme of reducing the required memory space that quantizes the confidence value into 256 intervals and represents it as one byte value from 0 to 255. This encoding reduces the required memory size to 152 kB (=606 $\times$ 256 LGP patterns $\times$ 1 byte).

We represent the computation time of our face detector as a linear function $T(t)=N\times t+C$, where N is the number of possible detection windows in the image, t is the average computation time to process one detection window, and C is a constant time that includes the image loading time, the preprocessing time (the time for transforming the input image into the MCT/LBP/LGP feature image in the case of MCT/LBP/LGP-based face detector, the time for making the integral image in the case of Viola–Jones face detector, the time for constructing the pyramid image) and the postprocessing time (the time for clustering or the time for EAM).

We measured the computation time of the Viola–Jones face detector [26], the MCT-based face detector [18], the LBP-based face detector [1], the proposed LGP-based face detector on a 2.83 GHz Intel Pentium IV PC system with 8 GB RAM and obtained the average computation time (Table 3) of several face detectors by averaging the total computation time of 10,000 320×240 input images. The average computation time of the Rowley–Baluja–Kanade face detector [22] and the Schneiderman–Kanade detector [37] was obtained from [26], which stated that their face detector was roughly 15 times faster than the Rowley–Baluja–Kanade face detector and roughly 600 times faster than the Schneiderman–Kanade face detector. The proposed LGB-based face detector is slightly slower than the MCT-based face detector and the LBP-based face detector due to the gradient computation for LGP feature transformation and the additional computation for EAM. However, the proposed LGB-based face detector is seven times faster than the Viola–Jones face detector because the proposed LGB-based face detector computes the weak classifier by one array reference to the lookup table, whereas the Viola–Jones face detector computes the weak classifier by more than six array references even with integral image.

6. Conclusion

Most existing face detection methods using LBP and MCT features are known to be relatively insensitive to globally changing intensity variations. However, they often fail to detect faces in which intensity varies locally due to local illumination, markup, wearing of glasses, and a variety of backgrounds. To overcome this sensitivity to local changing intensity variations, we proposed a novel face representation method called LGP, in which each bit has a value of 1 if the neighboring gradient of a given pixel is greater than the average of eight neighboring gradients, and 0 otherwise. Because LGP generated constant patterns irrespective of locally changing intensity variations along edges, it increased face detection accuracy drastically.

Furthermore, we proposed to use EAM, which accumulates evidence from multi-scale detection results and suppresses non-faces by thresholding the accumulated confidence score. EAM was very effective for reducing the number of false positives with a negligible extra computation time of accumulation because near a true face many detection windows with slightly different scales occur, which possess high confidence values, whereas near a non-face only a few detection windows may have high confidence values. EAM was also beneficial because it reduced the number of cascade stages required to achieve a small number of false positives.

Extensive experimental results validated the usefulness of the proposed LGP+EAM face detector. Results demonstrate that (1) LGP+EAM had the best face detection accuracy among many other face detection methods when tested on the MIT+CMU data set and the FDDB database, and improved the face detection rate by 5–27% depending on the difficulty of detection (face database), and (2) LGP+EAM less than half as many false positives as other face detection methods. Moreover, the proposed LGP+EAM was seven times faster than the Viola–Jones face detector and 100 times faster than the Rowley–Baluja–Kanade face detector. This high speed of face detection made the proposed face detector implementable on embedded systems. We also found that the proposed LGP+EAM method worked accurately on those digital media although the faces had many size variations, pose variations, illumination changes, racial variations, occlusions, and a tiny size of 20×22.

Finally, our proposed LGP+EAM face detection method has been already commercialized on several products: (a) face detectors on digital cameras, mobile phones and smart phones, (b) driver's drowsiness warning system for automobiles, (c) access control for

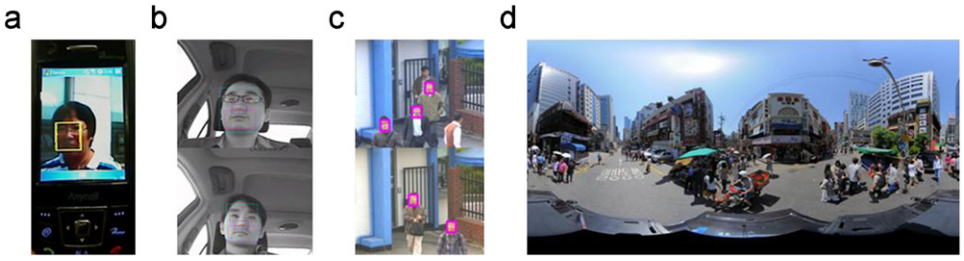

Fig. 16. Some commercial products using our face detection method.

CCTV surveillance systems, and (d) face removal for large-scale privacy protection system in street views (see Fig. 16).

Acknowledgments

This work was partially supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Core Technology Development for Breakthrough of Robot Vision Research support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C7000-1001-0006) and was partially supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2011-0027953).

References

[1] T. Ojala, M. Pietikainen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 29 (1) (1996) 51–59.
[2] H. Jin, Q. Liu, H. Lu, X. Tong, Face detection using improved lbp under Bayesian framework, in: Proceedings of the Third International Conference on Image and Graphics, 2004, pp. 306–309.
[3] L. Zhang, R. Chu, S. Xiang, S. Liao, S. Li, Face detection based on multi-block lbp representation, in: Proceedings of the Second International Conference on Biometrics, 2007, pp. 11–18.
[4] L. Zhang, R. Chu, S. Xiang, S. Liao, S. Li, Face detection based on multi-block lbp representation, in: Proceedings of Advances in Biometrics, 2007, pp. 11–18.
[5] T. Qian, R. Veldhuis, Illumination normalization based on simplified local binary patterns for a face verification system, in: Biometrics Symposium, 2007, pp. 1–6.
[6] S. Liao, X. Zhu, Z. Lei, L. Zhang, S. Li, Learning multi-scale block local binary patterns for face recognition, in: Proceedings of the Second International Conference on Biometrics, 2007, pp. 828–837.
[7] O. Lahdenoja, M. Laiho, A. Paasio, Reducing the feature vector length in local binary pattern based face recognition, in: Proceedings of the IEEE International Conference on Image Processing, 2005, pp. 11–14.
[8] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 2037–2041.
[9] L. Wolf, T. Hassner, Y. Taigman, Descriptor based methods in the wild, in: Faces in Real-Life Images Workshop in ECCV, 2008, pp. 1–14.
[10] C. Shan, S. Gong, P. McOwan, Facial expression recognition based on local binary patterns: a comprehensive study, Image and Vision Computing 27 (2009) 803–816.
[11] C. Frank, E. Noth, Automatic pixel selection for optimizing facial expression recognition using eigenfaces, in: Lecture Notes in Computer Science, 2003, pp. 378–385.
[12] X. Wang, T.X. Han, S. Yan, An hog-lbp human detector with partial occlusion handling, in: IEEE International Conference on Computer Vision, 2009, pp. 32–39.
[13] V. Kellokumpu, G. Zhao, S. Li, M. Pietikainen, Dynamic texture based gait recognition, in: Lecture Notes in Computer Science, 2009, pp. 1000–1009.
[14] V. Takala, T. Ahonen, M. Pietikainen, Block-based methods for image retrieval using local binary patterns, in: Proceedings of the 14th Scandinavian Conference on Image Analysis, 2005, pp. 882–891.
[15] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution grayscale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.
[16] M. Heikkila, M. Pietikainen, J. Heikkila, A texture-based method for detecting moving objects, in: Proceedings of the 15th British Machine Vision Conference, 2004, pp. 187–196.
[17] R. Zabih, J. Woodfill, Non-parametric local transforms for computing visual correspondence, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 151–158.
[18] B. Froba, A. Ernst, Face detection with the modified census transform, in: Proceedings of IEEE the Sixth International Conference on Face and Gesture Recognition, 2004, pp. 91–96.
[19] M. Turk, A. Pentland, Eigenface for recognition, Cognitive Neuro-science 3 (1) (1991) 70–86.
[20] E. Osuna, Support Vector Machines: Training and Applications, Ph.D. Thesis, MIT, EE/CS Dept., 1998.
[21] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (4) (2001) 349–361.
[22] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.
[23] H. Schneiderman, T. Kanade, A statistical method for 3D object detection applied to face and cars, in: Proceedings of Computer Vision and Pattern Recognition, 2000, pp. 746–751.
[24] A. Nefian, M. Hayes, Face detection and recognition using hidden Markov models, in: Proceedings of the IEEE International Conference on Image Processing, 1998, pp. 141–145.
[25] K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 39–51.
[26] P. Viola, M. Jones, Fast and robust classification using asymmetric adaboost and a detector cascade, in: Proceedings of Advances in Neural Information Processing System, 2001, pp. 1311–1318.
[27] S. Brubaker, J. Wu, J. Sun, M. Mullin, J. Rehg, On the design of cascades of boosted ensembles for face detection, International Journal of Computer Vision 77 (1) (2008) 65–86.
[28] C. Huang, H. Ai, Y. Li, S. Lao, High-performance rotation invariant multiview face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (4) (2007) 671–686.
[29] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[30] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, in: Machine Learning, 1999, pp. 80–91.
[31] Y. Rodriguez, Face Detection and Verification Using Local Binary Patterns, Ph.D. Thesis, Ecole Polytechnique Fédérale de Lausanne, 2006.
[32] C. Garcia, M. Delakis, Convolutional face finder: a neural architecture for fast and robust face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1408–1423.
[33] H.A. Rowley, Neural Network-based Face Detection, Ph.D. Thesis, Carnegie Mellon University, Pitsburgh, 1999.
[34] V. Jain, E. Learned-Miller, FDDB: A Benchmark for Face Detection in Unconstrained Settings, University of Massachusetts, Amherst, 2010.
[35] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, in: Proceedings of the European Conference on Computer, 2004, pp. 69–82.
[36] V. Subburaman, S. Marcel, Fast bounding box estimation based face detection, in: Proceedings of the ECCV Workshop on Face Detection: Where We Are, and What Next?, 2010.
[37] P. Viola, M. Jones, Robust real-time object detection, International Journal of Computer Vision 57 (2) (2002) 137–154.

Bongjin Jun received the B.S. degree in computer engineering from Busan National University, Korea, in 2000, and the M.S. and Ph.D. degree in computer engineering from Pohang University of Science and Technology (POSTECH), in 2002 and 2011, respectively. Now, he is working for Intelligent Media Lab., POSTECH as an Executive Manager. His research interests include computer vision, face analysis, and human analysis.

Daijin Kim received the B.S. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1981, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, 1984. In 1991, he received the Ph.D. degree in electrical and computer engineering from Syracuse University, Syracuse, NY.

During 1992–1999, he was an Associate Professor in the Department of Computer Engineering at DongA University, Pusan, Korea. He is currently a Professor in the Department of Computer Science and Engineering at POSTECH, Pohang, Korea. From 2010, he has been a Director of Pohang Institute of Intelligent Robotics (PIRO) and a director of Samsung Techwin-POSTECH Intelligent Media Research Center. His research interests include biometrics, human–computer interaction, and robotics.