

Accurate Face and Human Detection Using Hybrid Local Transform Features

Daijin Kim and Bongjin Jun

Abstract We propose two novel local transform features: local gradient patterns (LGP) and binary histograms of oriented gradients (BHOG). LGP assigns one if the neighboring gradient of a given pixel is greater than the average of eight neighboring gradients and zero otherwise, which makes the local intensity variations along the edge components robust. BHOG assigns one if the histogram bin has a higher value than the average value of the total histogram bins, and zero otherwise, which makes the feature computation time fast due to no further post-processing and SVM classification. We also propose a hybrid feature that combines several local transform features by AdaBoost feature selection method where the best local transform feature among several local transform features (LBP, LGP, and BHOG), which has the lowest classification error, is sequentially selected until we obtain the required classification performance. This hybridization makes the face and human detection robust to the global illumination change by LBP, the local intensity change by LGP, and the local pose change by BHOG, which improves the detection performance considerably. We apply the proposed local transform features and the hybrid feature to the face detection problem using MIT+CMU and FDDB face database and the human detection problem using INRIA human database. The experimental results show that the proposed LGP and BHOG features attain accurate detection performance and fast computation time, respectively, and the hybrid feature provides a considerable improvement of face detection and human detection.

Keywords Local binary pattern • Local gradient pattern • Binary histograms of oriented gradients • Feature hybridization • Face detection • Human detection

D. Kim (✉)

Department of Computer Science and Engineering, Pohang University of Science and Technology, San31, Hyoja-dong, Nam-gu, Gyeongbuk, Pohang 790-784, Korea
e-mail: dkim@postech.ac.kr

B. Jun

StradVision, Rm 106, Ji-Gok Research Building, San 31, Hyoja-dong, Nam-gu, Gyeongbuk, Pohang 790-784, Korea
e-mail: simple21@postech.ac.kr

1 Introduction

Face and Human detection is one of the important topics in computer vision. It has been widely used for the practical and real-time applications in many areas such as digital media (cell phone, smart phone, smart TV, digital camera), intelligent user interfaces (Wii, MS Kinect), intelligent visual surveillance, and interactive games. Conventional face and human detection methods usually take the pixel color (or intensity) [37] directly as the information cue. However, these cues are sensitive to the illumination changes and noises [29]. To tackle this problem, many researchers have introduced the transform features that convert the pixel color (or intensity) by a certain nonlinear transformation function. They can be categorized into two transform features: the intensity-based transform features and the gradient-based transform features.

First, the intensity-based transform features convert the pixel color (or intensity) into the encoded value by comparing the pixel value with the neighboring pixel value. Papageorgiou and Poggio [26] introduced the Haar-like features that encoded the differences in average intensities between two rectangular regions and they applied to extract the textures irrespective of pixel color (or intensity). Viola and Jones [39, 40] used the Haar-like features to detect the faces. They used an integral image [40] to compute the Haar-like features efficiently and an efficient scheme for constructing a strong classifier by cascading several weak classifiers using AdaBoost training. Yan et al. [41] proposed the binary Haar feature that kept only the directional relationship in the Haar feature computation. However, the discriminating power of a single binary Haar feature was too weak to construct a robust classifier. They also proposed the assembled binary Haar (ABH) feature that integrated three binary Haar features to improve the discriminative power of the binary Haar feature. However, the dimensionality of ABH feature is very huge. Furthermore, they proposed the locally assembled binary (LAB) Haar feature that combined 8 locally adjacent 2-rectangle to reduce the size of feature dimensionality. The LAB Haar feature represented the local intensity differences at various locations, scales, and orientations. Ojala et al. [24] proposed the local binary patterns (LBP) feature that was derived from a general definition of texture in a local neighborhood of the image. They encoded an image pixel into a 8-bit binary pattern that compared the intensity of center pixel within the 3×3 block with the intensity values of 8 boundary pixels with the 3×3 block and representing the comparison result as 1 or 0. One important advantage of the LBP feature was that it was invariant to the monotonic change of illumination. Zabin and Woodfill [42] proposed the census transform (CT) that is similar to the LBP feature. The LBP feature and its variants have been widely used in many applications: face detection [19, 43], face recognition [1, 44], facial expression recognition [12, 33], gender recognition [36], face authentication [16], gait recognition [21], image retrieval [38], texture classification [14, 25], shape localization [17], and object detection [15].

Second, the gradient-based transform features convert the pixel color (or intensity) into the gradient magnitude and orientation. Lowe [22] proposed the SIFT descriptor

that extracted distinctive invariant features from images and was invariant to image scale and rotation. The SIFT descriptor computed a histogram of local oriented gradients around the key point and represented the histogram in a 128 dimensional vector. It was obtained by computing the gradient magnitude and orientation on the key points, where the key points were obtained by finding the maxima and minima of the difference of Gaussian (DOG) images among three adjacent layers. It also required an image pyramid to make the SIFT descriptor scale invariant. Ke and Sukthankar [20] proposed the PCA-SIFT that used the principal component analysis (PCA) instead of histogram to normalize gradient patch. The feature vector was significantly smaller than the SIFT feature vector. They showed that PCA-based local descriptors were distinctive and robust to image deformations but it took a long computation time to extract the local descriptors. Bay et al. [2] proposed the speeded up robust features (SURF) that was an efficient implementation of SIFT by using the integral image. The SURF descriptor was obtained by computing the gradient magnitude and orientation on the key points, where the key points was obtained by finding the maxima of the Haar-like box filtered images. It did not require the image pyramid because it used many different sized box filters using integral image. Dalal and Triggs [4] proposed the histogram of oriented gradients (HOG) that divided the object into many fixed sized blocks, computed the HOG of each block, and represented the object by a concatenation of the block's HOG vectors. The HOG feature has been widely used in many applications: human detection [4, 6, 46], face recognition [5], object detection [10, 11] and emotion recognition [3]. Many researchers [9, 34, 45, 46] have also extended the original HOG to use variable-sized blocks, which improved the detection performance greatly.

In this chapter, we take two representative local transform features: local binary patterns (LBP) and histogram of oriented gradients (HOG) because LBP is robust to the global illumination change and HOG is robust to the local pose change. However, the local transform features have some limitations such that LBP is sensitive to local intensity changes due to makeup, wearing of glasses, and a variety of background and HOG requires a long processing time to compute the feature transformation.

To overcome these limitations, we propose two new local feature transforms: LGP and BHOG. LGP assigns one if the neighboring gradient of a given pixel is greater than the average of eight neighboring gradients, and zero otherwise, which makes the local intensity variations along the edge components robust. 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 face/face distance and face/non-face distance. BHOG assigns one if the histogram bin has a higher value than the average value of the total histogram bins, and zero otherwise, which makes the feature computation time fast due to no further post-processing and SVM classification.

We also propose a hybrid feature that combines several local transform features by AdaBoost feature selection method where the best local transform feature among several local transform features (LBP, LGP, and BHOG), which has the lowest classification error, is sequentially selected until we obtain the required

classification performance. This hybridization makes the face and human detection robust to the global illumination change by LBP, the local intensity change by LGP, and the local pose change by BHOG, which improves the detection performance considerably.

This chapter is organized as follows. Section 2 describes the LGP feature to overcome the limitation of the LBP feature. Section 3 describes the BHOG feature to speed up the computation of the HOG feature. Section 4 describes a hybridization of several local transform features that combines them by AdaBoost feature selection method. Section 5 describes the experimental results of face and human detection that demonstrates the usefulness of the proposed local transform features and the hybrid feature. Finally, Sect. 6 presents conclusions.

2 Local Gradient Patterns

Many variants of LBP have been applied to tasks such as face detection, face recognition, facial expression recognition, gender recognition, face authentication, gate recognition, image retrieval, texture classification, shape localization, and object detection. 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 generates many different patterns of local intensity variations and makes training of the face and human detection by AdaBoost difficult. To overcome this problem, we propose a novel face and human representation method called Local Gradient Patterns (LGP), which generates constant patterns irrespective of local intensity variations along edges.

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 1s and 0s into a binary code (See Fig. 1).

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 along on the circle (See Fig. 2). To obtain the values of pixel positions in the neighborhood for r and p , bilinear interpolation is necessary. It 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 $\bar{g} = \frac{1}{p} \sum_{n=0}^{p-1} g_n$. Then, $\text{LGP}_{p,r}(x_c, y_c)$ can be expressed as

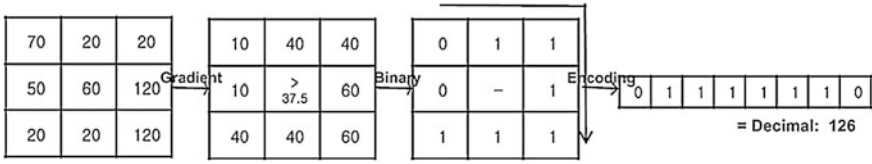


Fig. 1 The original LGP operator. © 2013 IEEE

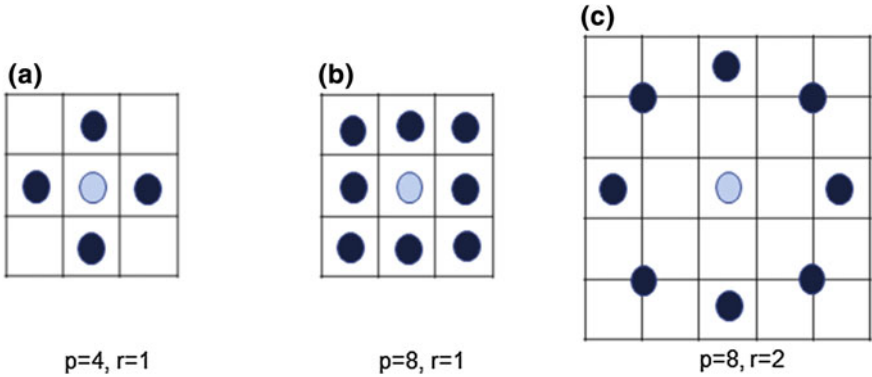


Fig. 2 Three examples of neighboring pixels: $LGP_{4,1}$, $LGP_{8,1}$ and $LGP_{8,2}$. © 2013 IEEE

$$LGP_{p,r}(x_c, y_c) = \sum_{n=0}^{p-1} s(g_n - \bar{g})2^n, \tag{1}$$

where

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

Figure 3 illustrates that LBP and LGP generate the same codes and the different codes depending on the global and local intensity changes. When the intensity levels of both the background and the foreground are changed together (globally), LGP and LBP both generate invariant patterns (See Fig. 3a). 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. 3b, c). This difference occurs because LGP generates patterns using the gradient difference ($s(g_n - \bar{g})$), whereas LBP generates patterns using the intensity difference ($s(i_n - i_c)$). For the nearly uniform color region, there exist the small variations of absolute intensity differences between two neighboring pixels. We can suppress these small variations of absolute intensity differences by setting the threshold as a predefined value that is a little greater than the average absolute difference.

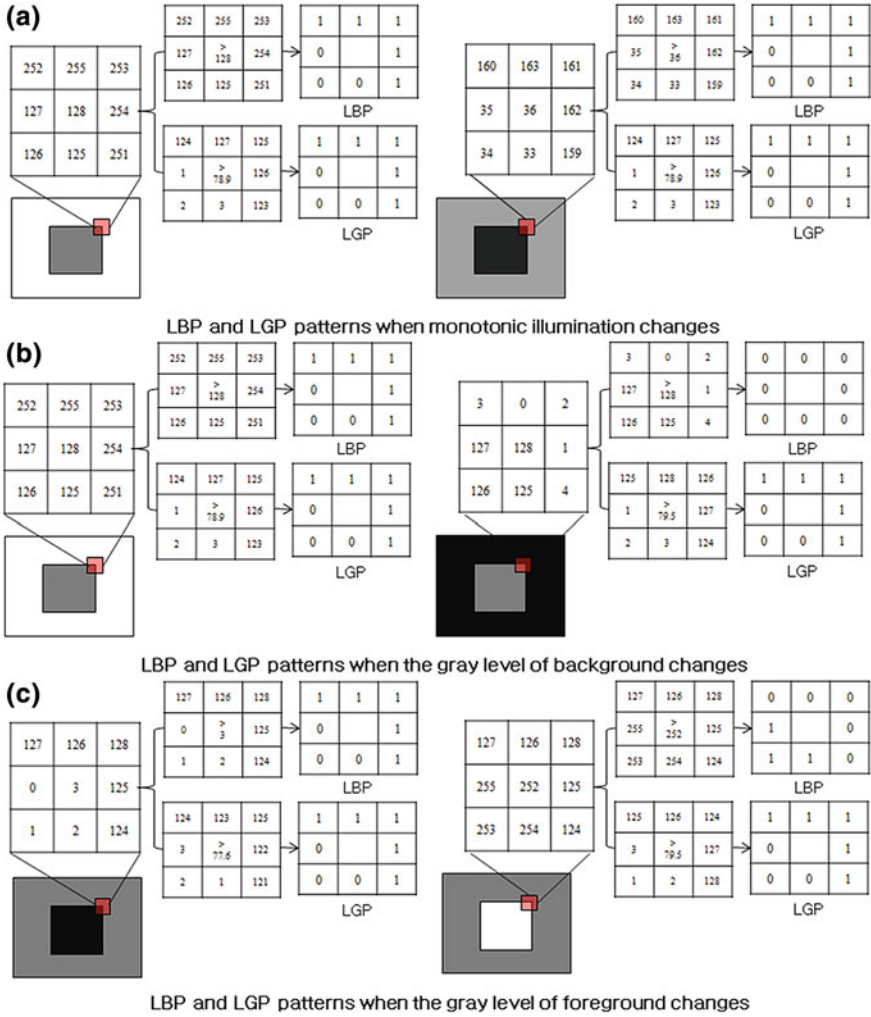


Fig. 3 LBP and LGP patterns when the intensity levels are changed globally or locally. © 2013 IEEE

3 Binary Histograms of Oriented Gradients

Dalal and Triggs [4] showed that the HOG feature combined with a linear SVM was a good detection performance of human beings. They took the overlapped block division method, the 1-D centered mask $[-1, 0, 1]$, and the L2-Hys normalization method. However, it showed a slow processing speed of 1 fps for the 320×240 image although it took very small number of search windows (800 windows per image).

Q. Zhu et al. [46] used a cascade of rejectors and AdaBoost training to select the features which needed to be evaluated in each stage. This method could process 320×240 images over the speed of 5 fps, while maintaining an accuracy level similar to the existing HOG methods. However, it was still not enough to run in real-time, because each HOG feature consisted of 36 dimensional histogram vectors for each block and the weak classifiers of AdaBoost were the linear SVMs with HOG features.

To overcome this problem, we propose a novel face and human representation method called the binary histograms of oriented gradients (BHOG) that assigns one if the histogram bin has a higher value than an average value of the total histogram bins and zero otherwise, where threshold is just . Therefore, the BHOG feature for a given block is represented by concatenating the binary 1s and 0s into a binary code (See Fig. 4). While the HOG feature represents each block by the 256 bit vector (8 bins \times 32 bits), the BHOG feature represents each block by the 8 bits, which makes the processing time efficient.

The BHOG feature is generated as follows. First, we compute the square of gradient magnitude and orientation of all pixels within the block. Second, we build the orientation histogram $HOG(b)$, $b = 0, 1, \dots, 7$ in the same way of generating the HOG feature. Third, we encode the orientation histogram into 8 bit vector, where each bit is determined by thresholding: If the histogram bin has a higher value than a given threshold, the 1 bit is assigned. Otherwise, the 0 bit is assigned. The decimal form of the 8 bit BHOG feature for a given block is expressed as

$$BHOG = \sum_{n=0}^7 s(HOG(n) - Th)2^n, \tag{3}$$

where Th denotes the average of HOG as $Th = \frac{1}{8} \sum_{n=0}^7 HOG(n)$ and a sign function $s(\cdot)$ is defined as

$$s(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

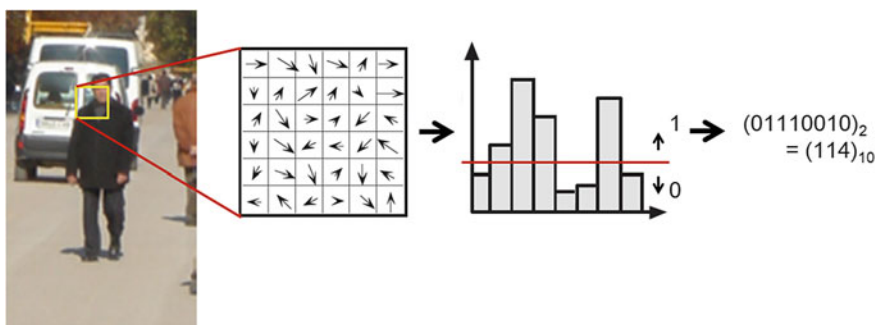


Fig. 4 Binary histograms of oriented gradients. © 2013 IEEE

The BHOG feature has several advantages over the HOG feature as follows. First, the BHOG feature does not require the square root operation in computing the gradient magnitude because it just compares the value of histogram bin with a given threshold. Second, the BHOG feature does not perform normalization of the orientation histograms which is the essential part in the original HOG, since it just requires the relative comparison between the value of histogram bin and a given threshold value. Third, the BHOG feature can be obtained by the AdaBoost training because it can be represented as one dimensional scalar value.

However, the HOG feature cannot use the Adaboost training because it is represented by a $N \times M$ dimensional vector that is obtained by concatenating N blocks, where each block is M dimensional vector. Therefore, the HOG feature is obtained by applying the linear SVM to the vector and then applying the Adaboost training to the scalar value of the SVM result. Finally, the BHOG feature uses the variable-sized blocks from 3×3 to $W \times H$, where W and H denote the width and height of the image, which can capture a lot of useful information that is spread over different scales and it can capture a large sized part of the human body (e.g. head, arm, leg).

4 Hybridization of Local Transform Features

We propose a hybridization of local transform features that combines them by AdaBoost feature selection method, where the best local transform feature among several local transform features (LBP, LGP, and BHOG), which has the lowest classification error, is sequentially selected until we obtain the required classification performance. The pool of feature candidates consists of a large set of point features in the case of LBP and LGP features and a huge number of block features with a variety of sizes from 3×3 to $W \times H$ in the case of BHOG feature. The selected features should not be redundant and characterize both intra-class variability and inter-class variability well. This hybridization makes the face and human detection robust to the global illumination change by LBP, the local intensity change by LGP, and the local pose change by BHOG, which improves the detection performance considerably. To select discriminative features from LBP, LGP, and BHOG, we use AdaBoost based on LBP, LGP, and BHOG.

The overall procedure of selecting the hybrid feature using the AdaBoost training is given below. First, we prepare the positive and the negative training images. Second, we initialize the weight values of the positive and the negative training images. Third, we obtain the positive and the negative training feature images of three different local transform features: LBP, LGP and BHOG. Fourth, we compute the classification errors for all feature images. Fifth, we select the best local transform feature that has the minimum classification error. Finally, we update the weight values of all the training images such that the training images incorrectly classified by the selected feature have large weight values and the training images correctly classified by the selected feature have small weight values in the

subsequent iterations. We prevent to re-select the previously selected feature by the other feature type by sharing the weight values among LBP, LGP and BHOG features.

After AdaBoost training, we obtain a strong classifier $H(\mathbf{C})$, where \mathbf{C} includes LBP, LGP, and BHOG feature images. Then, it is represented by the sum of weak classifiers as

$$H(\mathbf{C}) = \sum_{\mathbf{x} \in S_T^{\text{LBP}}} h_{\mathbf{x}}(\mathbf{L}(\mathbf{x})) + \sum_{\mathbf{x} \in S_T^{\text{LGP}}} h_{\mathbf{x}}(\mathbf{G}(\mathbf{x})) + \sum_{\mathbf{x} \in S_T^{\text{BHOG}}} h_{\mathbf{x}}(\mathbf{B}(\mathbf{IH}(\mathbf{x}))), \quad (5)$$

where \mathbf{L} is an LBP feature, \mathbf{G} is an LGP feature, \mathbf{IH} is an integral histogram [27] of the HOG feature whose size is $w \times h$ of one detection window, $\mathbf{B}(\cdot)$ is a binary HOG feature value computed from HOG feature vector, S_T^{LBP} , S_T^{LGP} , and S_T^{BHOG} are the sets of selected LBP, LGP and BHOG features at the final iteration, respectively, \mathbf{x} denotes the selected feature as $\mathbf{x} = (\text{type}, x, y, w, h)$ (If type is LBP or LGP, x and y represents feature location, while w and h has no meaning, if type is BHOG, x and y represent the center position of the selected block, while w and h represent the width and height of the selected block.), and $h_{\mathbf{x}}(\cdot)$ is the weak classifier that consists of a lookup table with a dimensionality of $2^N \{0, 2^N - 1\}$, N is bit length of LBP, LGP, and BHOG) whose index is just the LBP, LGP, or BHOG value.

The value at each index of the lookup table indicates that the smaller it is, the more positive training images have the index and the larger it is, the more negative training images have the index. The weak classifiers are constructed using AdaBoost training [13], which updates the weight of each training sample such that misclassified instances are given a higher weight in the subsequent iteration. Table 1 shows an overall procedure of selecting the hybrid feature using the AdaBoost training procedure and Table 2 shows a detailed sub-procedure of selecting the best feature.

5 Experimental Results and Discussion

5.1 Face Detection

5.1.1 Data Preparation

We prepared 30,000 images from the FDD06¹ database, which contained the faces with the race, illumination, color and texture variations. We detect the faces in the image manually and normalized the detected faces to the face images with a fixed size of 22×24 pixels using the manually marked both eye's center positions. We generated 300,000 training face images by shifting slightly the face images, scaling

¹See database(<http://imlab.postech.ac.kr/faceDB/FDD06/FDD06.html>).

Table 1 Hybrid feature selection using AdaBoost training. © 2013 IEEE

1. Prepare the training images $\{(T_i, c_i) i = 1, 2, \dots, N_p + N_n\}$, where N_p and N_n denote the number of positive and negative training images, respectively, $c_i = 0$ for $T_i \in P$ and $c_i = 1$ for $T_i \in N$, where P and N denote positive and negative training images, respectively.
2. Initialize the weights of the positive and negative training images as $w_i = \begin{cases} \frac{1}{N_p} & \text{for } c_i = 0, \\ \frac{1}{N_n} & \text{for } c_i = 1, \end{cases}$ define the set of selected features $S_1 = \{\}$, set the number of selected features to N_s , and set the values of the weak classifier $h_{\mathbf{x}_t}(\gamma) = 0$, where \mathbf{x}_t denotes one of LBP, LGP, and BHOG features, $t = 1, 2, \dots, N_s$ and the feature index $\gamma = 0, \dots, 2^N - 1$.
3. Apply LBP, LGP, and HOG to all positive and negative training images. Let L_i , G_i , and \mathbf{IH}_i be the positive and negative training LBP, LGP and integral histogram of HOG feature images, respectively.
4. For $t = 1, 2, \dots, T$
(a) Select the best feature \mathbf{x}_t with the classification error ϵ_{best_t} , by performing the tasks in Table 2.
(b) Update the weak classifier at the selected feature \mathbf{x}_t as $h_{\mathbf{x}_t}(\gamma) = h_{\mathbf{x}_t}(\gamma) + \alpha_t z_t(\gamma)$, where $\gamma = 0, \dots, 2^N - 1$ and $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
(c) Update the weights of positive and negative training images as if the type of \mathbf{x}_t is LBP, $w_{t+1}(i) = w_t(i) \cdot \begin{cases} e^{-\alpha_t}, & \text{if } z_t(L_i(\mathbf{x}_t)) = c_i, \\ e^{\alpha_t}, & \text{if } z_t(L_i(\mathbf{x}_t)) \neq c_i, \end{cases}$ if the type of \mathbf{x}_t is LGP, $w_{t+1}(i) = w_t(i) \cdot \begin{cases} e^{-\alpha_t}, & \text{if } z_t(G_i(\mathbf{x}_t)) = c_i, \\ e^{\alpha_t}, & \text{if } z_t(G_i(\mathbf{x}_t)) \neq c_i, \end{cases}$ if the type of \mathbf{x}_t is BHOG, $w_{t+1}(i) = w_t(i) \cdot \begin{cases} e^{-\alpha_t}, & \text{if } z_t(\mathbf{B}(\mathbf{IH}_i(\mathbf{x}_t))) = c_i, \\ e^{\alpha_t}, & \text{if } z_t(\mathbf{B}(\mathbf{IH}_i(\mathbf{x}_t))) \neq c_i, \end{cases}$
(d) Normalize the weights of positive and negative training images as $w_{t+1}(i) = \frac{w_{t+1}(i)}{\sum_{i=1}^{N_p+N_n} w_{t+1}(i)}$.
5. The final strong classifier is the sum of weak classifiers as $H(\mathbf{C}) = \sum_{\mathbf{x} \in S_T^{\text{LBP}}} h_{\mathbf{x}}(\mathbf{L}(\mathbf{x})) + \sum_{\mathbf{x} \in S_T^{\text{LGP}}} h_{\mathbf{x}}(\mathbf{G}(\mathbf{x})) + \sum_{\mathbf{x} \in S_T^{\text{BHOG}}} h_{\mathbf{x}}(\mathbf{B}(\mathbf{IH}(\mathbf{x})))$, where S_T^{LBP} , S_T^{LGP} , and S_T^{BHOG} are the set of selected feature positions at the final iteration T .

the face images with 0.95, 1.0, and 1.05 scale-factors, and rotating the face images by -15 , 0 , and 15 degrees in order to detect the faces irrespective of positions and scales. In addition, we mirrored the training face images to make them doubled. Figure 5 shows some typical training face images that were normalized by two eyes.

We prepared 17,000 non-face images from the FDD06 database, which did not contain the faces and generated 300,000 training non-face images by resizing the non-face images and taking the image patches with a fixed size of 22×24 pixels from the resized non-face images at random positions. These non-face images were used to train only the 1st stage of the cascade of face detectors, which will be explained later. From the 2nd stage of the cascade of face detectors, only the non-face images that were classified as false positives in the previous stage, were used to train the current stage face detector.

Table 2 A sub-procedure of selecting the best feature. 2013 IEEE

1. Generate the weight tables from the positive and negative training LBP, LGP, and BHOG feature images as $W_t^{k,\text{LBP}}(\mathbf{x}, \gamma) = \sum_{i,\mathbf{x},\gamma} w_i(i)I(L_i(\mathbf{x}) = \gamma)I(c_i = k)$, $W_t^{k,\text{LGP}}(\mathbf{x}, \gamma) = \sum_{i,\mathbf{x},\gamma} w_i(i)I(G_i(\mathbf{x}) = \gamma)I(c_i = k)$, $W_t^{k,\text{BHOG}}(\mathbf{x}, \gamma) = \sum_{i,\mathbf{x},\gamma} w_i(i)I(\mathbf{B}(\mathbf{H}_i(\mathbf{x})) = \gamma)I(c_i = k)$, where $k = 0$ or 1 for positive or negative training samples, respectively, and $I(\cdot)$ is an indicator function that takes a value of 1 if the argument is true, and 0 otherwise.

2. Compute the error $\varepsilon_t(\mathbf{x})$ for each lookup table as $\varepsilon_{\text{LBP}} = \sum_{\gamma} \min\{W_t^{0,\text{LBP}}(\mathbf{x}, \gamma), W_t^{1,\text{LBP}}(\mathbf{x}, \gamma)\}$, $\varepsilon_{\text{LGP}} = \sum_{\gamma} \min\{W_t^{0,\text{LGP}}(\mathbf{x}, \gamma), W_t^{1,\text{LGP}}(\mathbf{x}, \gamma)\}$, $\varepsilon_{\text{BHOG}} = \sum_{\gamma} \min\{W_t^{0,\text{BHOG}}(\mathbf{x}, \gamma), W_t^{1,\text{BHOG}}(\mathbf{x}, \gamma)\}$, $\varepsilon_t(\mathbf{x}) = \min\{\varepsilon_{\text{LBP}}, \varepsilon_{\text{LGP}}, \varepsilon_{\text{BHOG}}\}$.

3. Select the best feature position \mathbf{x}_t as $\mathbf{x}_t = \begin{cases} \mathbf{x} = \min_{\mathbf{x}} \varepsilon_t(\mathbf{x}), & \text{if } |S_t| < N_s, \\ \mathbf{x} = \min_{\mathbf{x} \in S_t} \varepsilon_t(\mathbf{x}), & \text{otherwise,} \end{cases}$

where N_s is the allowed number of selected feature positions.

4. Update the set of selected features as

if the type of \mathbf{x}_t is LBP, $S_{t+1}^{\text{LBP}} = \{S_t^{\text{LBP}} \cup \mathbf{x}_t\}$,

if the type of \mathbf{x}_t is LGP, $S_{t+1}^{\text{LGP}} = \{S_t^{\text{LGP}} \cup \mathbf{x}_t\}$,

if the type of \mathbf{x}_t is BHOG, $S_{t+1}^{\text{BHOG}} = \{S_t^{\text{BHOG}} \cup \mathbf{x}_t\}$, $S_{t+1} = \{S_{t+1}^{\text{LBP}} \cup S_{t+1}^{\text{LGP}} \cup S_{t+1}^{\text{BHOG}}\}$.

5. Determine the dominant class indicator $z_t(\gamma)$ of the feature value γ at the selected feature \mathbf{x}_t as $z_t(\gamma) = \begin{cases} 0, & \text{if } W_t^0(\mathbf{x}_t, \gamma) > W_t^1(\mathbf{x}_t, \gamma), \\ 1, & \text{otherwise.} \end{cases}$

**Fig. 5** Normalized training face images. © 2013 IEEE

We also prepared 15,000 images from the internet, which were not used for training and generated 150,000 validation face images in the same way of generating the training face images. We also prepared 15,000 non-face images from the internet, which were not used for training and generated 250,000 validation non-face images in the same way of generating the training non-face images.

5.2 Training Procedure

We have three different face detectors that use different features such as LBP, LGP and LBP+LGP+BHOG hybrid features, respectively. The AdaBoost training procedure of three face detectors is explained below.

First, we transform the training face and non-face images into the training face and non-face LBP, LGP, and BHOG feature images. Second, we compute the classification errors of all features. Third, we select one best feature with the minimum classification error at the current iteration. Fourth, we update the weight values of the training face and non-face feature images. Fifth, we check the stop condition that we achieve 99 % detection rate and 4 % false positive error rate using the validation face and non-face feature images. If the stop condition is satisfied, then we stop and obtain the selected features: the positions features in the case of LBP and LGP and the position and block features in the case of hybrid feature. Otherwise, we normalized the weight values of the training face and non-face feature images and go to the second step.

5.2.1 Cascade of Face Detectors

Since the proposed face detection method is based on classifying every possible window in the image as positive images or negative images, it takes long computation to detect the face in the high resolution image. To make the detection fast, we can employ the cascade of face detectors using the AdaBoost training method used by Viola and Jones [39].

In the real experiments, we trained three different cascades of face detectors using the LBP, LGP, and the hybrid feature images. However, we failed to train the cascade of face detectors using the BHOG feature images because the BHOG feature has only 8 different patterns in the case of 3×3 size of block. We set the maximum number of selected features of stage 1, 2, 3, and 4–26, 60, 120, and 400, respectively.

Figure 6 shows the selected features of three different cascade of face detectors using the LBP, LGP, and hybrid features, where white dots denote the positions of the selected point features in case of the LBP and LGP features and the center positions of the selected block features in the case of BHOG feature, and the rectangular boxes denote the sizes of the selected block features. We represent the center points of all the selected block features but did not represent the sizes of all the selected block features because it is very difficult to draw the boxes of all the selected block features within the face image. From Fig. 8, we know that (1) the LBP features are mostly selected from eye and mouth endpoints because they capture the common characteristics to all training face images, (2) the LGP features are widely selected from all face regions because they capture the locally changing gradient information and (3) the BHOG features are mostly selected from the eye, nose and mouth regions because they capture the common block information to all training face images.

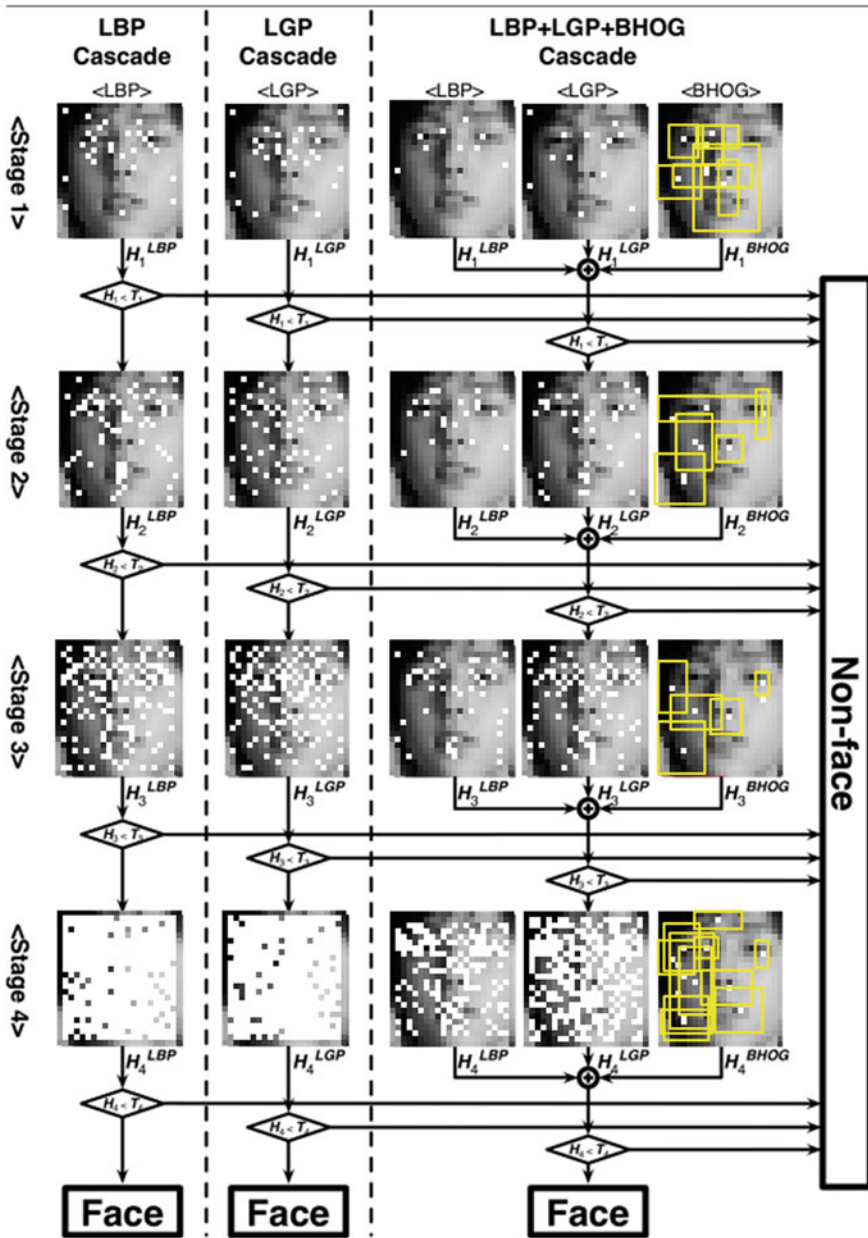


Fig. 6 Selected features of three cascades of face detectors. © 2013 IEEE

Table 3 shows the number of selected features in each stage that is determined from the training of the cascade of face detectors using the hybrid feature images. From Table 3, we know that (1) the LGP features are selected more than the LBP

Table 3 The number of selected features in each stage. 2013 IEEE

Feature	Stage 1	Stage 2	Stage 3	Stage 4	Total
# of LBP	8	19	39	146	212
# of LGP	12	34	76	242	364
# of BHOG	6	7	5	12	30
Total	26	60	120	400	606

Table 4 The computation time in each stage for training three different cascades of face detectors. 2013 IEEE

Cascade	Training time (LBP or LGP)	Training time (Hybrid)
Stage 1	≈1 min	≈6 min
Stage 2	≈5 min	≈40 min
Stage 3	≈30 min	≈4 h
Stage 4	≈23 h	≈3 days

and BHOG features because they are widely distributed over the all face region and (2) the BHOG features are rarely selected because they cover the large face components such as eyes, nose, and mouth.

Table 4 shows the computation time in each stage that is executed for the training of three different cascades of face detectors using the LBP, LGP and hybrid feature images, which run on the 2.83 GHz Intel Pentium IV PC system with 8 GB RAM. From Table 4, we know that the training time for the cascade of face detectors using the LBP and LGP feature images takes about one day while the training time for the cascade of face detectors using the hybrid feature images takes about four days.

5.2.2 Detection Performance

After training the proposed four-stage cascaded face detector, we evaluated the face detection accuracy using two kinds of face databases: the MIT+CMU database [30] (130 images with 483 faces), the Face Detection Data Set and Benchmark (FDDB²) database [18] (2,845 images with 5,171 faces). The face images in the MIT+CMU database are easy to detect because they are frontal and upright, and have mild illumination variations. The face images in the FDDB database are very difficult to detect because they include many occluded images and have large pose/illumination variations.

We considered six face detection methods for performance evaluation: the LBP feature-based face detector (LBP), the LGP feature-based face detector (LGP),

²See <http://vis-www.cs.umass.edu/fddb/results.html>.

the LBP+LGP feature-based face detector (LBP+LGP), the hybrid feature-based face detector (HYBRID). We compared four face detection methods (LBP, LGP, LBP+LGP, and HYBRID) with other existing face detection methods: Rowley-Baluja-Kanade [31], Viola-Jones [39], Mikolajaczyk et al. [23], Subburaman et al. [35].

Figure 7a, b show two receiver operating characteristic (ROC) curves that are obtained from several different face detection methods using the MIT+CMU database and the Fddb database, respectively. From Fig. 7a using the MIT+CMU database, we know that (1) the detection rate of the proposed HYBRID face detection method was the highest among all face detection methods by 0.959 when the false positive per image (FPPI) is one and (2) the number of false positives of the HYBRID, LBP+LGP, LGP, LBP, Viola-Jones, and Rowley-Baluja-Kanade methods at the 0.9 detection rate is 4, 7, 26, 67, 78, and 166, respectively,

From Fig. 7b using the Fddb database, we know that (1) the detection rates using the Fddb database are lower than those using the MIT+CMU database because the face images in the Fddb database has higher variations in the pose,

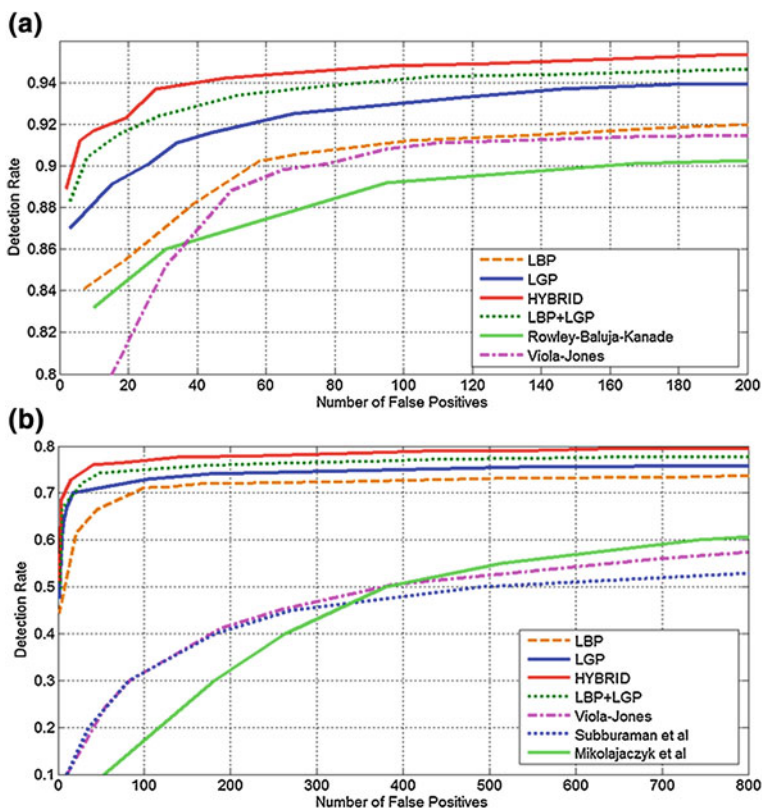


Fig. 7 ROC curves using (a) the MIT+CMU database and (b) the Fddb database. © 2013 IEEE

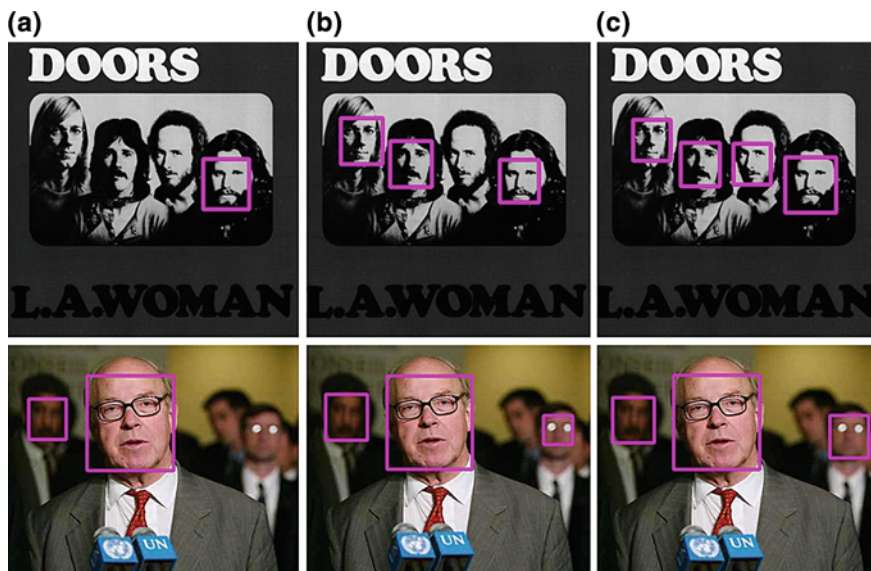


Fig. 8 Comparison of face detection results from (a) the LBP feature-based face detector, (b) the LGP feature-based face detector, and (c) the hybrid feature-based face detector. © 2013 IEEE

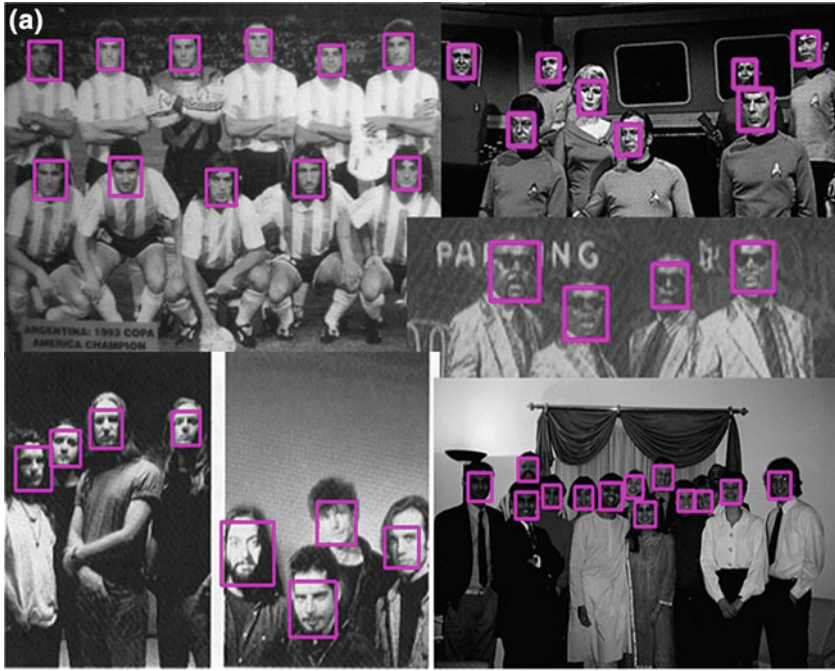
illumination, expression, and occlusion than those in the MIT+CMU database, (2) the detection rate of the proposed HYBRID method was the highest among all face detection methods by 78.9 % when the false positive per image (FPPI) is 0.1, (3) the detection rate of the HYBRID, LBP+LGP, LGP, LBP, Viola-Jones, Mikolajczyk et al., and Subburaman et al. methods at the 0.1 FPPI are 78.2, 76.3, 74.2, 72.1, 46.2, 45.6, and 42.3 %, respectively.

Figure 8 shows the face detection results using the MIT+CMU database (top row) and the Fddb database (bottom row), where (a), (b) and (c) are obtained from the LBP feature-based face detector, the LGP feature-based face detector and the hybrid feature-based face detector, respectively. From Fig. 8, we know that the HYBRID feature-based face detector succeeds to find most of faces, even tiny faces with a size of 22×24 , but the LBP and LGP feature-based face detectors fail to find them occasionally.

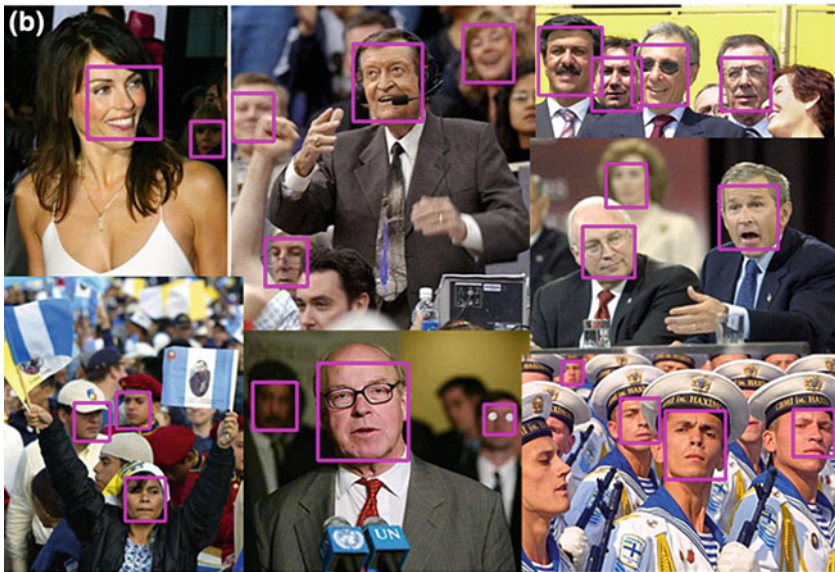
Figure 9 shows several face detection results using the hybrid feature-based face detector on the MIT+CMU and Fddb database, respectively.

5.2.3 Memory Size

Each weak classifier must store the confidence value at each LBP, LGP, and BHOG value in the lookup table, where the confidence value is represented by a real number, which consists of 8 bytes. Therefore, each weak classifier requires a memory space of 2,048 bytes (= 256 LGP patterns \times 8 bytes). Because stages 1–4



Face detection results using the MIT+CMU database



Face detection results using the Fddb database

Fig. 9 Face detection results. 2013 IEEE

consist of 26, 60, 120 and 400 weak classifiers respectively, the total required memory space is 1.2 Mbytes (= $606 \times 2,048$ 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 Kbytes (= 606×256 LGP patterns \times 1 byte).

5.2.4 Computation Time

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 LBP, LGP, BHOG feature image, the time for making integral histogram of HOG in the case of the hybrid-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).

We measured the computation time on a 2.83 GHz Intel Pentium IV PC system with 8 GB RAM. Table 5 shows the preprocessing time and the average computation time of several face detectors, where it is the average of the computation time of 10,000 320×240 input images.

The average computation times of the Rowley-Baluja-Kanade face detector [31] and the Schneiderman-Kanade face detector [39] were referred from [39], 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 LGP feature-based face detector is slightly slower than the LBP-based face detector due to the gradient computation for LGP feature transformation. However, the LGP feature-based face detector is seven times faster than the Viola-Jones face detector because the LGP feature-based

Table 5 Comparison of average computation time among several face detectors (unit: 10^{-3} s). 2013 IEEE

Detector	Pyramid	Feature	Face	Total
	Image	Transform	Detection	Time
LBP feature-based [24]	1.7	1.76	6.20	9.66
LGP feature-based	1.7	2.05	6.07	10.12
HYBRID feature-based	1.7	9.78	25.78	37.26
Viola-Jones [39]	0.0	0.16	70.06	70.22
Rowley-Baluja-Kanade [31]	–	–	–	1053.3
Schneiderman-Kanade [32]	–	–	–	42132.0

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.

The proposed hybrid feature-based face detector is roughly 2 times faster than the Viola-Jones face detector. Since most of the features of hybrid feature-based face detector consist of LBP and LGP features, there are a few number of BHOG features. Accordingly, hybrid feature-based face detector requires a few number of integral histogram computations which take much computation time. In contrast, all the weak classifiers of Viola-Jones face detector consist of Haar-like features which require high number of integral image computation.

5.3 *Human Detection*

5.3.1 **Data Preparation**

We prepared 618 images from the INRIA database [4], which contained 1,208 humans with the pose, illumination, appearance, and occlusion variations. We detect the human in the image manually and normalized the detected humans to the human images with a fixed size of 32×64 pixels using the manually marked head and toe positions. We generated 59,180 training human images by shifting slightly the human images and scaling the human images with 0.95, 1.0, and 1.05 scale-factors in order to detect the humans irrespective of positions and scales. In addition, we mirrored the training human images to make them doubled. Figure 10 shows some typical training human images that were normalized by the head and toe.

We prepared 1,218 nonhuman images from the INRIA database [4], which did not contain humans and generated 100,000 training nonhuman images by bootstrapping and resizing the nonhuman images and taking the image patches with a fixed size of 32×64 pixels from the resized nonhuman images at random positions. These nonhuman images were used to train only the first stage of the cascade of human detectors. From the 2nd stage of the cascade of human detectors, only the nonhuman images that were classified as false positives in the previous stage, were used to train the current stage human detector.

5.3.2 **Training Procedure**

We have two different human detectors that use different features such as BHOG and LBP+LGP+BHOG hybrid features, respectively. The BHOG feature uses the variable size of blocks from 4×4 to $W \times H$, where W and H denote the width and height of the window image, which it can capture a lot of useful information that is spread over different scales and it can capture a large sized part of the human body



Fig. 10 Normalized training human images. © 2013 IEEE

(e.g. head, arm, leg). The AdaBoost training procedure of two human detectors is explained below.

First, we transform the training human and nonhuman images into the training human and nonhuman LBP, LGP, and BHOG feature images. Second, we compute the classification errors of all feature images. Third, we select one best feature with the minimum classification error at the current iteration. Fourth, we update the weight values of the training human and nonhuman feature images. Fifth, we check the stop condition that we achieve 96 % detection rate and 8 % false positive error rate using the validation human and nonhuman feature images. If the stop condition is satisfied, then we stop and obtain the selected features: the position features in the case of LBP and LGP and the position and block features in the case of hybrid feature. Otherwise, we normalize the weight values of the training human and nonhumane feature images and go to the second step.

5.3.3 Cascade of Human Detectors

We also take the cascade of human detectors to make the human detection fast. In real experiments, we trained two different cascades of human detectors using BHOG and LBP+LGP+BHOG hybrid feature images because the LBP and LGP features failed to train the human detectors. We set the maximum number of selected features of stage 1, 2, 3, 4, and 5–40, 80, 160, 320, and 1,600, respectively.

Figure 11 shows the selected features of two different cascade of human detectors using the BHOG and hybrid features, where white dots denote the

positions of the selected point features in the case of the LBP and LGP features and the center positions of the selected block features in the case of BHOG feature, and the rectangular boxes denote the sizes of the selected block features. We represent the center points of all the selected block features but did not represent the sizes of all the selected block features because it is very difficult to draw the boxes of all the selected block features. From Fig. 11, we know that (1) the LBP features are mostly selected from the shoulder because they capture the common characteristics to all training human images, (2) the LGP features are mostly selected from the arms and legs with high variations because they capture the locally changing gradient information, (3) the BHOG features are widely selected from all-human regions such as head, arms, legs, and torso because they capture the common block information to all training human images.

Table 6 shows the number of selected features in each stage that is determined from the training of the cascade of human detectors using the hybrid feature images. From Table 6, we know that (1) the LGP features are selected more than the LBP features because they are widely distributed over the all-human region and (2) the BHOG features are most widely selected over the whole body region because they cover the large body part components such as arms, legs and torso.

Table 7 shows the training time of two different cascade of human detectors using the BHOG and hybrid features, which runs on the 2.83 GHz Intel Pentium IV PC system with 8 GB RAM. From Table 7, we know that the training of the cascade of human detectors using the BHOG feature images takes about seven days while the training of the cascade of human detectors using the hybrid feature images takes about nine days.

5.3.4 Detection Performance

After training the proposed five-stage cascaded human detector, we evaluated the human detection accuracy using the INRIA database [4] that contained 288 test images with 1,132 humans.

We considered four human detection methods for performance evaluation: the BHOG feature-based human detector (BHOG), the LGP+BHOG feature-based human detector (LGP+BHOG) the hybrid feature-based human detector (HYBRID). We compared three human detection methods (BHOG, LGP+BHOG, and HYBRID) with other existing human detection methods: HOG [4] and VJ (Viola-Jones) [7] using the evaluation protocol based on Pascal measure [8].

Figure 12 shows the receiver operating characteristic (ROC) curve that is obtained from several different human detection methods using the INRIA database. From Fig. 12, we know that (1) the detection rate of the HYBRID, LGP+BHOG, BHOG, HOG_{64×128}, HOG_{32×64}, and VJ at the one false positive rate per images (FPPI) was 85.5, 83.5, 79.5, 78.9, 41, and 58 %, respectively, which means that the proposed HYBRID human detection method was the highest among all other human detection methods, and (2) the number of false positives of the

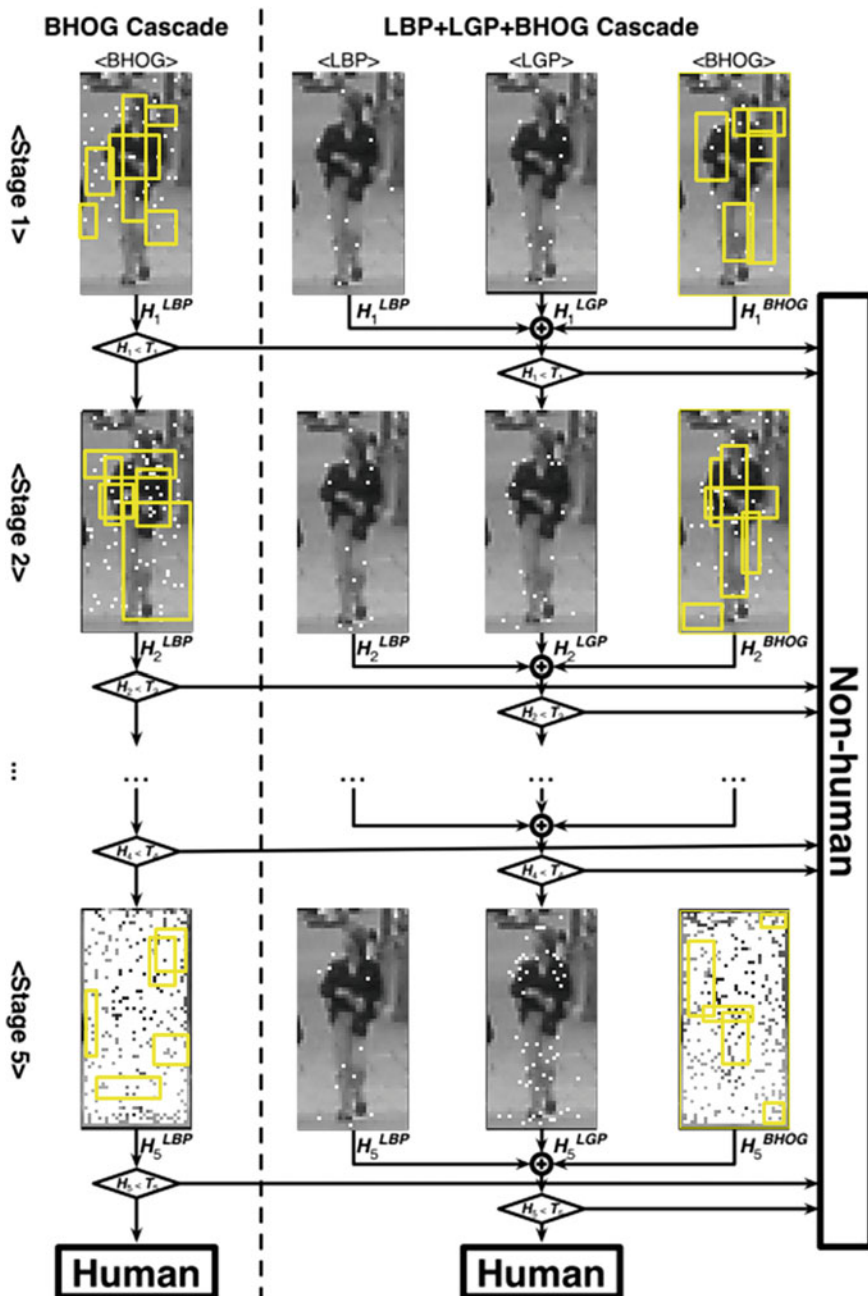


Fig. 11 Selected features of two cascades of human detectors. © 2013 IEEE

Table 6 The number of selected features in each stage. 2013 IEEE

Feature	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Total
# of LBP	8	12	17	16	12	65
# of LGP	14	18	31	43	54	160
# of BHOG	18	50	112	261	1,534	1,975
Total	40	80	160	320	1,600	2,200

Table 7 The training time of two different cascades of human detectors. 2013 IEEE

Cascade	Training time (BHOG)	Training time (Hybrid)
1 stage	≈8 min	≈10 min
2 stage	≈30 min	≈50 min
3 stage	≈4 h	≈4 h
4 stage	≈1 day	≈2 days
5 stage	≈5 days	≈6 days

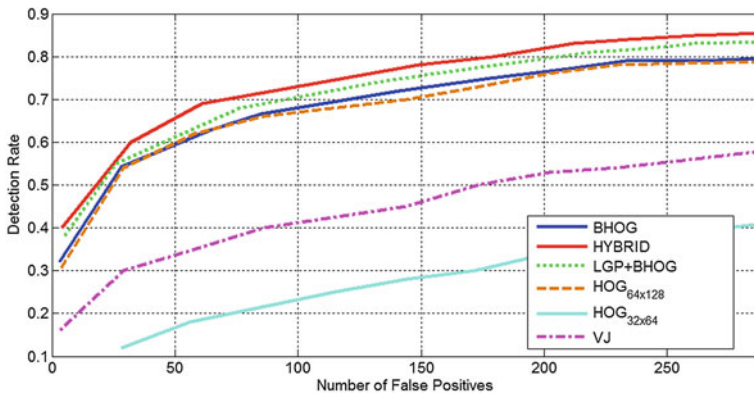


Fig. 12 ROC curves using the INRIA database. © 2013 IEEE

HYBRID, LGP+BHOG, BHOG, and HOG at the 70 % detection is 70, 92, 120, and 145, respectively.

Figure 13 shows the human detection results using the INRIA database, where (a), (b) and (c) are obtained from from the HOG-based human detector, the BHOG-based human detector, and the hybrid feature-based human detector, respectively. From Fig. 13, we know that the HYBRID feature-based human detector succeeds to find most of humans even small sized human with a size of 32×64 , but the HYBRID and HOG-based human detectors fail to find them occasionally.

Figure 14 shows several human detection results using the hybrid feature-based human detector on the INRIA database.

Fig. 13 Comparison of human detection results. © 2013 IEEE



We also evaluated the human detection accuracy using the MIT-CBCL³ database that contained 924 front/back-view positive images (no negative images). Instead of training on the MIT-CBCL database, we use our trained detectors on the INRIA database and tested them on the MIT-CBCL database. We achieve that (1) the detection rate of the HYBRID, LGP+BHOG, BHOG, and HOG at the zero false positive rate per images (FPPI) was 93.1, 92.6, 90.2, and 84.5 %, respectively, which means that the proposed HYBRID human detection method was the highest among all other human detection methods, and (2) this indicates that our detectors have good generalization performance.

5.3.5 Computation Time

We measured the computation time of the HOG human detector [4], the proposed BHOG-based human detector, and the proposed hybrid-based human detector on a 2.83 GHz Intel Pentium IV PC system with 8 GB RAM. Table 8 shows the average

³See <http://cbcl.mit.edu/software-datasets/PedestrianData.html>.



Fig. 14 Human detection results using the INRIA database. © 2013 IEEE

computation time of two human detectors, where it is the average of computation time of 1,000 320×240 input images. From Table 8, we know that (1) the existing HOG-based human detector works slowly in that it takes about $490 \cdot 10^{-3}$ s (≈ 2 fps) and the BHOG-based human detector works fast in that it takes about $52 \cdot 10^{-3}$ s (≈ 20 fps), which implies that the proposed BHOG-based human detector is about

Table 8 Comparison of average computation time among several human detectors (unit: 10^{-3} s). 2013 IEEE

Detector	Pyramid	Feature	Human	Total
	Image	Transform	Detection	Time
HOG feature-based	1.7	87	401.3	490
BHOG feature-based	–	6.01	46	52.01
Hybrid feature-based	1.7	9.78	165.78	177.26
Cascade+HOG [46]	–	–	–	214
GPU implementation of HOG [28]	–	–	–	19

10 times faster than the existing HOG-based human detector, and (2) the hybrid feature-based human detector is roughly three times slower than the BHOG-based human detector because it uses the hybrid features. One interesting point is that the BHOG-based human detector shows 1 % higher detection rate than the HOG-based human detector in spite of its faster computation time.

6 Conclusion

The most commonly used face and human detection method was local transform feature-based method. Many researchers have introduced many different approaches using local transform features: specifically local binary patterns (LBP) and histograms of oriented gradients (HOG). Each approach had its own advantage in that LBP was robust to monotonic illumination variations and HOG was robust to local pose variations. However, these methods have some limitations such that LBP was sensitive to locally changing intensity changes and HOG required a huge computation time for the feature transformation.

To overcome the limitations of the previous approaches, we proposed two novel local feature transformation methods: local gradient patterns (LGP) and binary HOG (BHOG) and proposed a hybridization of local transform features that combined several local features (LBP, LGP, and BHOG or HOG) by AdaBoost feature selection method to improve the face and human detection performance given below.

LGP encoded an image pixel into a 8-bit binary pattern by comparing the gradient of the given pixel and the average of its 8 neighboring gradients. It was invariant to the local gradient variations that were caused by makeup, wearing of glasses, and a variety of background, and had higher discriminant power than LBP.

BHOG binarized the histogram values of HOG by thresholding them with the average value of the total histogram bins. It did not require the square root operation in computing the gradient magnitude and the normalization of the orientation histograms because it just compared the value of histogram bin with a given

threshold and enabled to obtain the face and human detectors by the AdaBoost training because it was represented as one dimensional scalar value.

The hybridization of the multiple local transform features selected relevant features from the feature pool of LBP, LGP, and BHOG in order to improve the detection performance considerably. It took advantages of each local transform feature: LBP's robustness to local illumination change, LGP's robustness to locally changing intensity, and BHOG's robustness to local pose change.

We applied the proposed local transform features and its hybridization to face and human detection to validate the usefulness of the proposed methods. First, the face detection rates of LBP, LGP and the hybridization of LBP, LGP, and BHOG features using MIT+CMU database were 90, 93, and 96 %, respectively, which showed that the LGP feature resulted in better face detection rate than the LBP feature, and the hybrid feature resulted in the best face detection rate among them. Second, the human detection rates of HOG, BHOG and the hybridization of LBP, LGP and BHOG features using INRIA database were 79, 80, and 86 %, respectively, which showed that BHOG feature had similar detection rate but 10 times faster than HOG feature and the hybrid feature resulted in the best human detection rate among them. From all the results, we can conclude that the proposed local transform features and its hybrid feature are very effective for the face and human detection rate in terms of the performance and operating speed.

Acknowledgements This work is supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as the Global Frontier Project.

References

1. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. *IEEE Trans Pattern Anal Mach Intell* 28(12):2037–2041
2. Bay H, Ess A, Tuytelaars T, Gool LV (2008) SURF: speeded up robust features. *Comput Vis Image Underst* 110(3):346–359
3. Dahmane M, Meunier J (2011) Emotion recognition using dynamic grid-based HoG features. In: *Proceedings of IEEE international conference on automatic face and gesture recognition*, pp 884–888
4. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 886–893
5. Deniza O, Buenaño G, Salido J (2011) Face recognition using histograms of oriented gradients. *Pattern Recogn Lett* 32(12):1598–1603
6. Dollar P, Belongie S, Perona P (2010) The fastest pedestrian detector in the west. In: *Proceedings of the British machine vision conference*, pp 1–11
7. Dollar P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: a benchmark. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 304–311
8. Dollar P, Wojek C, Schiele B, Perona P (2011) Pedestrian detection: an evaluation of the state of the art. *IEEE Trans Pattern Anal Mach Intell* 34(4):743–761
9. Enzweiler M, Gavrilu DM (2009) Monocular pedestrian detection: survey and experiments. *IEEE Trans Pattern Anal Mach Intell* 31(12):2179–2195

10. Felzenszwalb P, Girshick R, McAllester D (2010) Cascade object detection with deformable part models. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 2241–2248
11. Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. *IEEE Trans Pattern Anal Mach Intell* 32(9):1627–1645
12. Feng X, Pietikainen M, Hadid A (2005) Facial expression recognition with local binary patterns and linear programming. *Pattern Recognit Image Anal* 15(2):546–548
13. Froba B, Ernst A (2004) Face detection with the modified census transform. In: Proceedings of IEEE international conference on automatic face and gesture recognition, pp 91–96
14. Grimes DB, Rao RPN (2003) A bilinear model for sparse coding. *Neural Inf Process Syst* 15:1287–1294
15. Heikkilä M, Pietikainen M, Heikkilä J (2004) A texture-based method for detecting moving objects. In: Proceedings of British machine vision conference, pp 187–196
16. Heusch G, Rodriguez Y, Marcel S (2006) Local binary patterns as an image preprocessing for face authentication. In: Proceedings of international conference on automatic face and gesture recognition, pp 9–14
17. Huang X, Li SZ, Wang Y (2004) Shape localization based on statistical method using extended local binary pattern. In Proceedings of international conference on image and graphics, pp 184–187
18. Jain V, Miller EL (2010) Fddb: a benchmark for face detection in unconstrained settings. University of Massachusetts, Amherst
19. Jin H, Liu Q, Lu H, Tong X (2004) Face detection using improved LBP under Bayesian framework. In: Proceedings of international conference on image and graphics, pp 306–309
20. Ke Y, Sukthankar R (2004) PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 511–517
21. Kellokumpu V, Zhao G, Li S, Pietikainen M (2009) Dynamic texture based gait recognition. In: Proceedings of international conference on biometrics, pp 1000–1009
22. Lowe DG (2004) Distinctive image features from scale invariant keypoints. *Int J Comput Vision* 60(2):91–110
23. Mikolajczyk K, Schmid C, Zisserman A (2004) Human detection based on a probabilistic assembly of robust part detectors. In: Proceedings of the European conference on computer vision, pp 69–82
24. Ojala T, Pietikainen M, Harwood D (1996) A comparative study of texture measures with classification based on feature distributions. *Pattern Recogn* 29(1):51–59
25. Ojala T, Pietikainen M, Maenpää T (2002) Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 24(7):971–987
26. Papageorgiou C, Poggio T (2000) CA trainable system for object detection. *Int J Comput Vision* 38(1):15–33
27. Porkili F (2005) Integral histogram: a fast way to extract histograms in cartesian spaces. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 829–836
28. Prisacariu V, Reid I (2009) FastHOG—a real-time GPU implementation of HOG. Department of Engineering Science, Oxford University
29. Randen T, Husoy JH (1999) Filtering for texture classification: a comparative study. *IEEE Trans Pattern Anal Mach Intell* 21(4):291–310
30. Rowley HA (1999) Neural network-based face detection. Ph.D. thesis, Carnegie Mellon University, Pittsburgh
31. Rowley H, Baluja S, Kanade T (1998) Neural network-based face detection. *IEEE Trans Pattern Anal Mach Intell* 20(1):23–38
32. Schneiderman H, Kanade T (2000) A statistical method for 3D object detection applied to faces and cars. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 746–751

33. Shan C, Gong S, McOwan P (2009) Facial expression recognition based on local binary patterns: a comprehensive study. *Image Vis Comput* 27:803–816
34. Shet VD, Neumann J, Ramesh V, Davis LS (2007) Bilattice-based logical reasoning for human detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8
35. Subburaman V, Marcel S (2010) Fast bounding box estimation based face detection. In: *Proceedings of ECCV workshop on face detection: where we are and what next?*
36. Sun N, Zheng W, Sun C, Zou C, Zhao L (2006) Gender classification based on boosting local binary pattern. In: *Proceedings of international symposium on neural networks*, pp 194–201
37. Swain M, Ballard D (1991) Color indexing. *Int J Comput Vision* 7(1):11–32
38. Takala V, Ahonen T, Pietikainen M (2005) Block-based methods for image retrieval using local binary patterns. In: *Proceedings of Scandinavian conference on image analysis*, pp 882–891
39. Viola P, Jones M (2004) Robust real-time face detection. *Int J Comput Vision* 57(2):137–154
40. Viola P, Jones M, Snow D (2005) Detecting pedestrians using patterns of motion and appearance. *Int J Comput Vision* 63(2):153–161
41. Yan S, Shan S, Chen X, Gao W (2008) Locally assembled binary (LAB) feature with feature-centric cascade for fast and accurate face detection. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 1–7
42. Zabih R, Woodfill J (1994) Non-parametric local transforms for computing visual correspondence. In: *Proceedings of European conference on computer vision*, pp 151–158
43. Zhang L, Chu R, Xiang S, Liao S, Li S (2007) Face detection based on multi-block LBP representation. In: *Proceedings of international conference on biometrics*, pp 11–18
44. Zhang W, Shan S, Gao W, Chen X, Zhang H (2005) Local gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. In *Proceedings of IEEE international conference on computer vision*, pp 786–791
45. Zhang L, Wu B, Nevatia R (2007) Detection and tracking of multiple humans with extensive pose articulation. In: *Proceedings of IEEE international conference on computer vision*, pp 1–8
46. Zhu Q, Avidan S, Yeh M, Cheng K (2006) Fast human detection using a cascade of histograms of oriented gradients. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 1491–1498