Complexity Reduced Face Detection Using Probability-Based Face Mask Prefiltering and Pixel-Based Hierarchical-Feature Adaboosting

Jing-Ming Guo, *Senior Member, IEEE*, Chen-Chi Lin, Min-Feng Wu, Che-Hao Chang, and Hua Lee, *Fellow, IEEE*

***Abstract*—The Adaboosting has attracted attention for its efficient face-detection performance. However, in the training process, the large number of possible Haar-like features in a standard sub-window becomes time consuming, which makes specific environment feature adaptation extremely difficult. This letter presents a two-stage hybrid face detection scheme using Probability-based Face Mask Pre-Filtering (PFMPF) and the Pixel-Based Hierarchical-Feature Adaboosting (PBHFA) method to effectively solve the above-mentioned problems in cascade Adaboosting. The two stages both provide far less training time than that of the cascade Adaboosting and thus reduce the computation complexity in face-detection tasks. In particular, the proposed PFMPF can effectively filter out more than 85% nonface in an image and the remaining few face candidates are then secondly filtered with a single PBHF Adaboost strong classifier. Given a $\mathrm{M} \times \mathrm{N}$ sub-window, the number of possible PBH features is simplified down to a level less than $\mathrm{M} \times \mathrm{N}$, which significantly reduces the length of the training period by a factor of 1500. Moreover, when the two-stage hybrid face detection scheme are employed for practical face-detection tasks, the complexity is still lower than that of the integral-image based approach in the traditional Adaboosting method. Experimental results obtained using the gray feret database show that the proposed two-stage hybrid face detection scheme is significantly more effective than Haar-like features.**

***Index Terms*—Adaboost, face detection, hierarchical feature, probability-based face mask.**

I. Introduction

FACE detection has become an important research topic for the direct potential applications such as biometrics, human-computer interfaces, and surveillance. Recently, the Rectangle-Feature (RF) based boosted detector was proposed by Viola and Jones [1]. RFs are local image features and the variation can be significant, depending on the focused local regions. Viola utilized a variant of Adaboost [2] for RF selection to extract a small subset of all possible RFs form the improvement of detection results.

The recent research has made substantial improvement on two aspects of the face-detection problem. One was on the selection of effective features and the other one was on the classifier designation with effective learning algorithm. Recently, a new set of rotated Haar-like features was presented by Lienhart [3], which can improve the accuracy for the cases of rotated face. Rowley *et al.*, also developed a neural network system for facial detection [4]. In addition, comparisons have been made about training time under different feature numbers. However, the practical detection time cannot be improved comparing to the typical Adaboost.

In this letter, a two-stage hybrid face detection scheme is proposed for significantly reducing training time and detection time to improve the feasibility of specific environment feature adaptation in face detection. Each method has a different advantage: The first stage, PFMPF, can filter out most nonface and thus accurately reserve face candidates for the second stage, PBHF Adaboost strong classifier.

The rest of this work is organized as follows. The proposed PFMPF is introduced in Section II. Section III describes the proposed pixel-based hierarchical-feature. Experimental results are presented in Section IV, and Section V draws the conclusions.

Manuscript received November 30, 2010; revised March 26, 2011; accepted April 01, 2011. Date of publication April 25, 2011; date of current version June 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitrios Tzovaras.

J.-M. Guo, C.-C. Lin, M.-F. Wu, and C.-H. Chang are with the Department of Electrical Engineering, National Taiwan University of Science and Technology, 10607 Taipei, Taiwan (e-mail: jmguo@seed.net.tw; m9707324@mail.ntust.edu.tw; m9607208@mail.ntust.edu.tw; jasonchang11@hotmail.com).

H. Lee is with the Department of Electrical and Computer Engineering University of California, Santa Barbara, CA 93106-9560 USA (e-mail: hualee@ece.ucsb.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2011.2146772

II. Proposed Probability-Based Face Mask Pre-Filtering

In this section, the proposed Probability-based Face Mask Pre-Filtering (PFMPF) is presented to reserve the face candidates with a higher accuracy.

A. Generation of Probability-Based Convolutional Face Mask

The two main concepts of the first stage, PFMPF, are that pixel-based face mask and its probability table (also called histogram table). Some previous approaches endeavor to generate an excellent pixel-based classifier, yet they only take the original pixel values as input. Instead, we create a convolutional face mask detection strategy with pixel values, the corresponding probability, and standard deviation as inputs. The details of the probability-based face mask generation are presented as follows.

Algorithm: Generation of Probability-based Convolutional Face Mask

Input: K training positive images X_i of size $M \times N$ (example:24 × 24). $i = 1 \ldots K$

Output: 1. Probability table $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{GreyValue})$ of each position of a positive face train image set.

2. Initial $M \times N$ weights (initial face mask).

begin

1. Gather statistics of the pixel grey level probability

First, initialize $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{GreyValue}) = 0$

for each pixel position $0 \leq \boldsymbol{x} < 24, 0 \leq \boldsymbol{y} < 24$

for each positive image $\boldsymbol{X}_i$.

$$\boldsymbol{P}(\boldsymbol{x},\ \boldsymbol{y},\ \boldsymbol{X_i}(\boldsymbol{x},\ \boldsymbol{y}))^{\leftarrow} \boldsymbol{P}(\boldsymbol{x},\ \boldsymbol{y},\ \boldsymbol{X_i}(\boldsymbol{x},\ \boldsymbol{y}))^{+1}$$

A probability table (histogram table) of positive images is generated.

2. Next, generate the initial convolutional face mask.

a. Calculate the mean value $\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{y})$ of each position (x, y) of all the positive images $\boldsymbol{X_i}$.

b. Calculate the standard deviation $\boldsymbol{D}(\boldsymbol{x}, \boldsymbol{y})$ of each position (x, y).

for $0 \leq \boldsymbol{x} < 24, 0 \leq \boldsymbol{y} < 24$

$$\boldsymbol{D}(\boldsymbol{x},\ \boldsymbol{y}) = \sqrt{\frac{\sum_{\boldsymbol{i}=1}^{\boldsymbol{K}} (\boldsymbol{X_i}(\boldsymbol{x},\ \boldsymbol{y}) - \boldsymbol{M}(\boldsymbol{x},\ \boldsymbol{y}))^2}{\boldsymbol{K}}}$$

c. The initial weights of the face mask are created.

for $0 \leq \boldsymbol{x} < 24, 0 \leq \boldsymbol{y} < 24$

$$\boldsymbol{W}(\boldsymbol{x},\ \boldsymbol{y}) = \frac{1}{\boldsymbol{D}(\boldsymbol{x},\ \boldsymbol{y})}$$

end

The histogram table is particularly generated for all of the face samples. The standard deviation is to measure the dynamic range of the pixel values in a specific position. A low standard deviation indicates the pixel values in position (x, y) tend to be very close, and vice versa. The reason that this strategy works is because normally in a face block, the pixel values of some specific positions are nearly in some specific ranges, such as the positions of eyes, eyebrows and cheek. Thus the standard deviation provides an excellent filter to remove impossible candidates with high dynamic fluctuations at these regions. Here below we demonstrate how to exploit the generated histogram table and the face mask to determine whether an unknown $\mathrm{M} \times \mathrm{N}$ block is a face or not. Given a candidate image region of size 24×24, the detection steps as organized as below:

Compute the inner product between the probability table and the face mask by

$$\boldsymbol{Weight\_Sum} = \sum_{\boldsymbol{x}=0}^{24} \sum_{\boldsymbol{y}=0}^{24} \quad \boldsymbol{P}(\boldsymbol{x},\ \boldsymbol{y},\ \boldsymbol{X_i}(\boldsymbol{x},\ \boldsymbol{y})) \bullet \boldsymbol{W}(\boldsymbol{x},\ \boldsymbol{y}). \tag{1}$$

For most 24×24 positive samples, the $\boldsymbol{Weight\_Sum}$ is closer to a higher fixed value $\boldsymbol{High\_Weight\_Sum}$. Relatively, for most 24×24 negatives sample, $\boldsymbol{Weight\_Sum}$ will be closer to a lower fixed value $\boldsymbol{Low\_Weight\_Sum}$. Consequently, if a threshold is set to classify the face and nonface candidates, it can provide good classified results using the initial weights:

$$classify = \begin{cases} face, & if\ Weight\_Sum \geq \text{ threshold} \\ nonface, & otherwise. \end{cases} \tag{2}$$

For test samples, the detection rate and false positive rate are around 95.35% and 5.68%, respectively.

III. Pixel-Based Hierarchical-Feature Adaboosting

In this section, the proposed pixel-based hierarchical-feature Adaboosting [8] is described. RFs [1] are local image features and the possible variations can be huge, depending on the considered local regions. Conversely, the PBH features employ statistics manner to generate features, and which is yielded from the global face information. Thus, the possible variations of the PBH features is much less than that of the PBH feature, and which can significantly reduce the training time. The PBH feature-generation procedure and detection strategies are organized in details.

A. Generation of Pixel-Based Hierarchical Features

Because the core problem of the Haar-like features is that most of the features are not good candidates for practical use, it is highly desirable to produce an alternative set of candidate features for the typical face structure. To achieve this goal, the PBH features are selected with the following procedure. And it is noted that the PBH features, once produced, can be applied to most cases.

1) First, input training positive images X_i of size $\mathrm{M} \times \mathrm{N}$.
2) The average value of each training image X_i is T_i, which is used as the threshold for corresponding training image X_i to produce the thresholded binary image B_i. An example is shown in Fig. 1(a).
3) Given all the B_i, calculate the probability of black pixel occurrence at each position (x, y) and obtain $P(x, y)$. Subsequently, the order table $O(x, y)$ is created according to the probability $P(x, y)$ distribution. Enter value $(1 \sim \mathrm{M} \times \mathrm{N})$ into $O(x, y)$ with the probability value in position (x, y). This means, when the probability of position (x, y) is the maximum, set $O(x, y) = 1$, and $O(x, y) = 2$ when the probability of position (x, y) is the second highest, and so forth. An example is shown in Fig. 1(b).
4) All possible features $\boldsymbol{F}_j$ and the corresponding feature values in a $\mathrm{M} \times \mathrm{N}$ sub-window can be produced according to the order table $O(x, y)$. First, the summation of the pixel values in the training image is calculated and denoted as sum. The features value $\boldsymbol{F}_j$ is computed as

$$F_j = sum - \left(\sum_{k=1}^{k=j} X_i(x, y),\ where\ O(x, y) = k \right)$$

where $X_i(x, y)$ denotes the pixel value of the training image X_i at position (x, y).

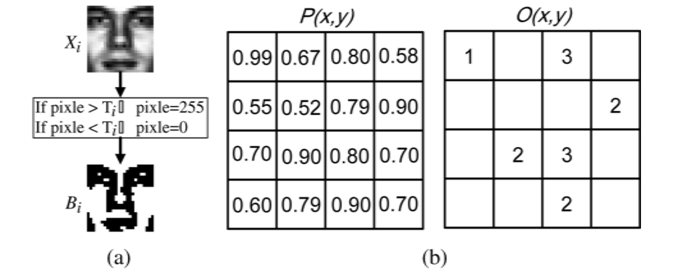

Fig. 1. Example of PBH features selection. (a) Step 2 to produce binary image. (b) Step 3 to produce the order table.

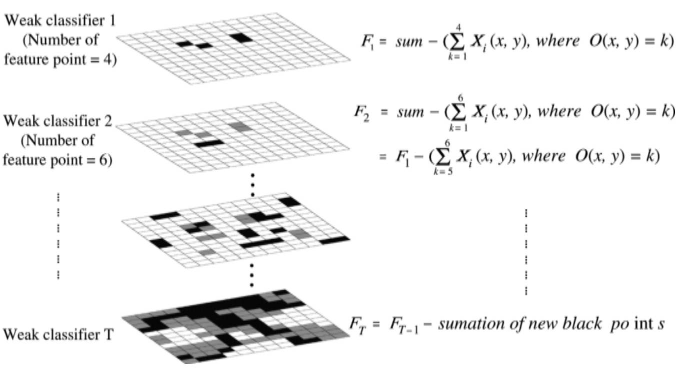

Fig. 2. Strong classifier equivalent concept using PBH features..

5) Finally, these features and training image X_i are fed into the Adabootst algorithm. Similar to the Haar-like features, each PBH feature can be considered as a weak classifier. Through the Adaboost algorithm, the threshold θ_j and inequality p_j can be derived to classify face and nonface candidates.

The main difference between pixel-based hierarchical (PBH) features and Haar-like features is that the features employed in PBH is obtained by subtracting overall pixel values in black region from the whole pixel values in a sub-window. The numbers of all possible features in $\mathrm{M} \times \mathrm{N}$ image are not over $\mathrm{M} \times \mathrm{N}$. Nevertheless, these features are best that have excellent ability to classify. The main point is that the training time is thus extremely shorter than that of Haar-like features and is able to achieve acceptable detection ability.

B. Detection Strategies

The construction of a strong classifier is detailed in [1]. The strong classifier can be obtained by combining the selected weak classifiers from the Adaboosting method and reorganized, as in Fig. 2.

In addition, the majority voting [4] is adopted to delete some false positives. Because a face in a test image is generally repeatedly detected at close positions, which means when a face is detected at position $(\boldsymbol{x}, \boldsymbol{y})$, it is more likely to be detected over the region $(\boldsymbol{x} \pm \boldsymbol{r}, \boldsymbol{y} \pm \boldsymbol{r})$, where the term r is a tunable variable in practical applications. Consequently, a sub-window at the position $(\boldsymbol{x}, \boldsymbol{y})$ is considered as a nonface, if there are no multiple positively detected results from voting over the region $(\boldsymbol{x} \pm \boldsymbol{r}, \boldsymbol{y} \pm \boldsymbol{r})$. In average this scheme can filter out 2.3785 nonface candidates with FERET database with $\boldsymbol{r}$ is set at 20.

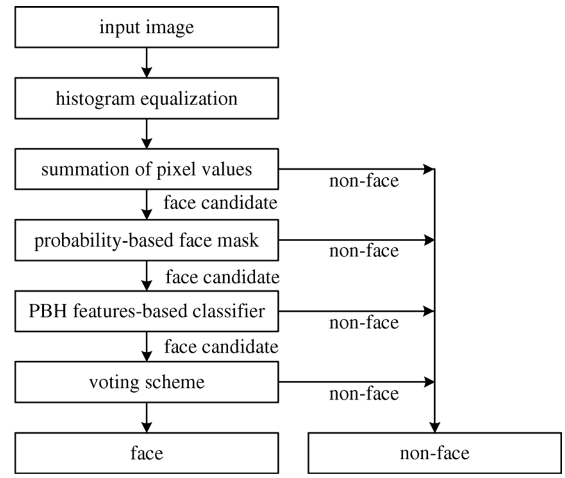

Fig. 3. Overall proposed face detection flow chart.

Finally, we wish to point out that an added procedure is used in this work to improve the detection efficiency. Generally, histogram equalization is utilized as a preprocessing step to enhance inadequate lighting conditions. In addition, we found that pixel summation of a facial variation is generally bounded in a specific interval of $[a, b]$. By setting the values of a and b properly, many candidate regions can be eliminated from the process before the detection process is conducted and the detection efficiency is improved as a result. To provide a clear concept of the proposed scheme, the overall proposed face detection flow chart is illustrated in Fig. 3.

IV. Experimental Results

In this section, the proposed two-stage hybrid face detection scheme is applied to practical face detection to demonstrate its capability and performance.

In the first experiment, the MIT-CBCL face database [7] is employed for training. The image size in MIT-CBCL face database is 19×19, and which is rescaled to 24×24 before trained.

The gray feret fa and fb database [5], [6] and CMU database [9] are employed to compare the performances between the proposed method and that uses Haar-like features. Those face databases include various variations, such as slight rotation (in-plane and out-of-plane), variation in facial expression, poor illumination, various clothes patterns, and different races. Table I shows the performance comparisons. The main objective of using this database is to test different variations on a face image. Notably, both of the methods use 200 features to yield a fair comparison.

The performance of a face detection system is normally evaluated based four metrics: 1) training time, 2) detection time, 3) detection rate, and 4) false positive rate. Although the detection rate and false positive rate are somewhat inferior to that of Haar-like feature, the difference on the training time is huge between the proposed scheme and that adopts Haar-like features. When Haar-like features are used, it takes more than 10 days to complete the training process, while it takes only 10 minutes by PBH features. This discrepancy is due to the number of exploited features. There are more than 160 000 for Haar-like features and less than 576 for PBH features. Although the difference in number is simply by a factor of ~300, the practical

TABLE I
PERFORMANCE COMPARISONS BETWEEN TRADITIONAL ADABOOSTING AND THE PROPOSED METHOD USING FERET AND CMU DATABASE

	Database	Detection rate	False positives rate
Traditional Adaboosting	FERET	98.70968%	6.535×10^{-8}
Propose method	FERET	98.0645%	2.9407×10^{-7}
Traditional Adaboosting	CMU	92.5926%	6.91387×10^{-6}
Proposed method	CMU	90.7407%	1.27802×10^{-5}

TABLE II
COMPUTATION COMPLEXITY COMPARISON BETWEEN HARR-LIKE FEATURES AND THE PROPOSED METHOD (SUBBPOSE SUB-WINDOW OF SIZE $\mathrm{M} \times \mathrm{N}$)

<table>
<tr><td>Feature type / Operation number</td><td colspan="2">Haar-like features</td><td colspan="2">Proposed two-stage hybrid scheme</td></tr>
<tr><td rowspan="2">Addition/ Subtraction</td><td>Integral Image</td><td>3×(M×N)</td><td>PFMPF</td><td>M×N</td></tr>
<tr><td>K weak classifier</td><td>(suppose all are 2-rectangle feature) 7×K</td><td>PBH feature</td><td>Less than 2×M×N</td></tr>
<tr><td>Addition/ Subtraction (Suppose sub-indow of size 24×24)</td><td colspan="2">Case1: 1784 (suppose 8 features are 2-rectangle features)
Case2: >1784 (suppose 3-rectangle or 4-rectagle features are used)</td><td colspan="2">85% are 576 and 15% are less than 1728</td></tr>
</table>

difference in training time is around by a factor of $\sim$1500. The reason is that too many Haar-like features cause data swapping between memory and hard drive, while the proposed PBH features can be solely put in the memory, and thus which causes the huge difference in training time.

Table II documents the computation complexity comparisons of these two methods when applied to practical detection. The upper part shows the complexity proportional to a sub-window size, $\mathrm{M} \times \mathrm{N}$. The complexity of using Haar-like features is further divided into integral image evaluation and K weak classifiers value calculation. Since the histogram equalization is applied to each sub-window to ease the lighting effect, each sub-window has to compute its corresponding integral image. So the complexity of computing the integral image is $\mathrm{O}(\mathrm{M} * \mathrm{N})$. The lower part of Table II shows the complexity when $\mathrm{M} \times \mathrm{N} = 24 \times 24$. The best scenario by [1] using Haar-like features is 1784 addition/subtraction when all the $K (= 8)$ weak classifiers are 2-rectangle features. However, when 3-rectangle or 4-rectangle features are involved, the number of operations is higher than 1784.

An example using the majority voting strategy is illustrated in Fig. 4. The five images on the left are corresponding to the scaled images from the original image with the scaling factor $\alpha = 1.25\mathrm{k}$. As it can be seen, all the test faces have been successfully detected for $\mathrm{k} = 0, \ldots, 4$. When $\mathrm{k} = 3$ and $\mathrm{k} = 4$, two

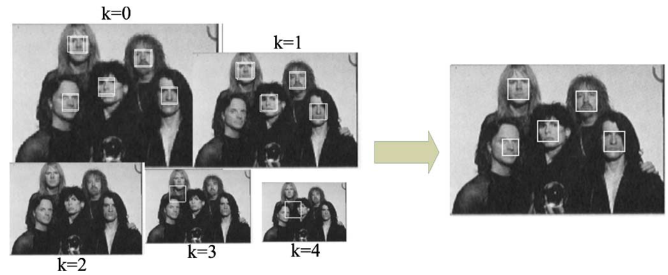

Fig. 4. An example of using majority voting strategy. (k denotes the power of the scaling factor $\alpha = 1.25\mathrm{k}$)..

false-positive detections are involved. For both cases, the voting numbers are one, which means the cases can be eliminated from the process. Good results have been obtained, as shown on the right of Fig. 4, with the majority voting strategy.

Finally, test cases with MIT-CBCL face database have been used to calculate the summation statistics. The horizontal axis represents the summation of pixel values and the vertical axis is the probability of summation. When the parameters a and b are set at 70 000 and 80 000, respectively. Approximately, 1.78965% of the nonface candidates are eliminated, which gives slight improvement of detection efficiency.

V. CONCLUSIONS

In this letter, a hybrid face detector with the Probability-based Convolutional Face Mask (PFMPF) and the Pixel-Based Hierarchical-Feature Adaboosting (PBHFA) is proposed. The goal is to improve the processing efficiency for the classification of face and nonface candidates. As documented in the experimental results, the training time is significantly reduced, and the detection rate remains competitive to the traditional Adaboosting method. In summary, the proposed method is an effective approach and the implementation is straightforward. The simplicity and computation efficiency make this approach an excellent candidate for real-time surveillance system.

REFERENCES

[1] P. Viola and M. J. Jones, “Robust real-time object detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[2] Y. Freund and R. Schapire, “A decision theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, 1997.
[3] R. Lienhart, A. Kuranov, and V. Pisarevsky, “Empirical analysis of detection cascades of boosted classifiers for rapid object,” in *DAGM 25th Pattern Recognition Symp.*, 2003, pp. 297–304.
[4] H. A. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, 1998.
[5] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, “The FERET database and evaluation procedure for face recognition algorithms,” *J. Image Vis. Comput.*, vol. 16, no. 5, pp. 295–306, 1998.
[6] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face recognition algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 1090–1104, 2000.
[7] CBCL Face Database #1 MIT Center for Biological and Computation Learning [Online]. Available: http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html
[8] J.-M. Guo and M.-F. Wu, “Pixel-based hierarchical-feature face detection,” in *IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, March 2010, pp. 1638–1641.
[9] CMU Frontal Face Images [Online]. Available: http://vasc.ri.cmu.edu//idb/html/face/frontal_images/index.html