

# Real-Time Face Detection and Recognition in Complex Background

Xin Zhang, Thomas Gonnot, Jafar Saniie

Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA

Email: xzhang78@hawk.iit.edu, tgonnot@hawk.iit.edu, saniie@iit.edu

**How to cite this paper:** Zhang, X., Gonnot, T. and Saniie, J. (2017) Real-Time Face Detection and Recognition in Complex Background. *Journal of Signal and Information Processing*, 8, 99-112.  
<https://doi.org/10.4236/jsip.2017.82007>

**Received:** March 25, 2017

**Accepted:** May 16, 2017

**Published:** May 19, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

This paper provides efficient and robust algorithms for real-time face detection and recognition in complex backgrounds. The algorithms are implemented using a series of signal processing methods including Ada Boost, cascade classifier, Local Binary Pattern (LBP), Haar-like feature, facial image pre-processing and Principal Component Analysis (PCA). The Ada Boost algorithm is implemented in a cascade classifier to train the face and eye detectors with robust detection accuracy. The LBP descriptor is utilized to extract facial features for fast face detection. The eye detection algorithm reduces the false face detection rate. The detected facial image is then processed to correct the orientation and increase the contrast, therefore, maintains high facial recognition accuracy. Finally, the PCA algorithm is used to recognize faces efficiently. Large databases with faces and non-faces images are used to train and validate face detection and facial recognition algorithms. The algorithms achieve an overall true-positive rate of 98.8% for face detection and 99.2% for correct facial recognition.

## Keywords

Face Detection, Facial Recognition, Ada Boost Algorithm, Cascade Classifier, Local Binary Pattern, Haar-Like Features, Principal Component Analysis

---

## 1. Introduction

Real-time face detection and facial recognition play an important role in applications such as robot intelligence, smart cameras, security monitoring or even criminal identification. Conventional algorithms for face detection and facial recognition are designed for still-face images or color images. In color images, the colors increase data complexity by mapping pixels onto a high-dimensional space, which greatly reduces the processing speed and accuracy of the face detection and recognition [1].

There are several approaches towards facial recognition problems. Given the fact, the faces are usually round or oval with same color, one simplest approach is to use the color segmentation to detect faces. However, using color segmentation is not able to adapt to the changing environment, such as lighting conditions. More adaptive and robust methods may not be able to operate in real time since they require more computational power. Moreover, adaptive algorithms usually employ statistical concepts in various degrees, such as template matching [2], Support Vector Machine (SVM) [3], color segmentation [4] or neural network [5]. More reliable descriptors such as Histogram of Oriented Gradient (HOG) [6], Scale-Invariant Feature Transform (SIFT) [7], Local Binary Pattern (LBP) [8], or Haar-like features [9] are used to determine facial features for face detection. The facial recognition is based on Principal Component Analysis (PCA) [10], Linear Discriminant Analysis (LDA) [11], holistic matching method [12] and feature-based method [7]. For practical applications, the faces need to be detected and recognized in real-time and often in complex backgrounds.

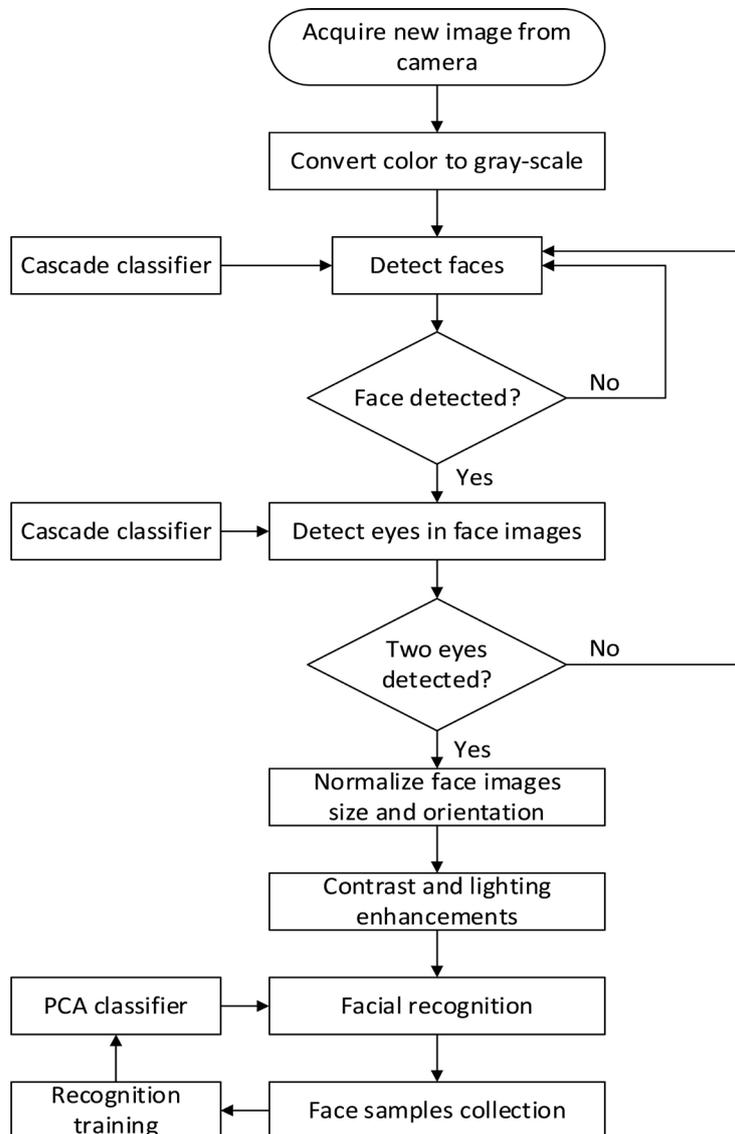
The algorithms proposed in this paper process gray-scale images to detect and recognize faces in real-time with high accuracy. The combination of Ada Boost algorithm and the cascade classifier [13] improves the detection accuracy. The face detection algorithm uses a cascade classifier based on the  $LBP_{(8,2)}^{u2}$  descriptor [8], providing a higher processing speed. The eye detection also uses a cascade classifier but based on the Haar-like descriptor to ensure low false-positive face detection rate. The result of facial recognition training can be improved significantly through an efficient pre-processing on training data. After training, the PCA algorithm is used for the facial recognition. The flowchart for real-time face detection and recognition is shown in **Figure 1**.

The implemented algorithm can be segmented into three stages: 1) Faces and eyes detection; 2) Facial images normalization and enhancement, and 3) Facial recognition and face sample collection. In stage 1, two different cascade classifiers are used to detect the faces and the eyes respectively. The training process of these two classifiers is done by the Ada Boost algorithm. In stage 2, faces detected in previous stage are normalized to a fixed size and orientation. In this stage, the backgrounds are discarded; the contrast and lighting get enhanced. In stage 3, the algorithm tracks the differences of faces in detection windows. In the case of significant difference, the algorithm will recognize the face using PCA and collect it to train the recognition algorithm further. With the help of pre-processing and eye detection module, the method proposed in this paper can operate more accurately regardless of the background.

## 2. Descriptors for Real-Time Detection

### 2.1. $LBP_{(8,2)}^{u2}$ Descriptor

The  $LBP_{(8,2)}^{u2}$  descriptor is used to extract facial features for the face detection. LBP stands for Local Binary Pattern, and every pattern of the facial image is encoded and counted to construct the spatially enhanced histogram representing local primitives. The subscript of  $LBP_{(8,2)}^{u2}$  indicates the LBP descriptor is using



**Figure 1.** Flowchart for real-time face detection and recognition.

8 sampling points within a radius of 2 pixels. The  $u2$  superscript indicates that the descriptor is using uniform patterns. This descriptor uses 58 bins to include 58 uniform patterns and 1 bin to include 198 non-uniform patterns. Uniform patterns account for almost 90% of the local primitives [14] and there are two transitions from 0 - 1 or 1 - 0 in each 8-bit binary number at most. Due to the shorter length of the histogram, the calculation can be greatly simplified by using the  $LBP_{(8,2)}^{u2}$  descriptor. Each sample histogram is compared with the template histogram to find the threshold for each region. The encoding process of  $LBP_{(8,2)}^{u2}$  descriptor is shown in **Figure 2**.

## 2.2. Haar-Like Descriptor

The Haar-like descriptor is utilized to extract eye features. Each Haar-like feature is composed of neighboring rectangular regions, which are shown in **Figure 3**. Haar-like features have multiple neighboring rectangular regions. The values

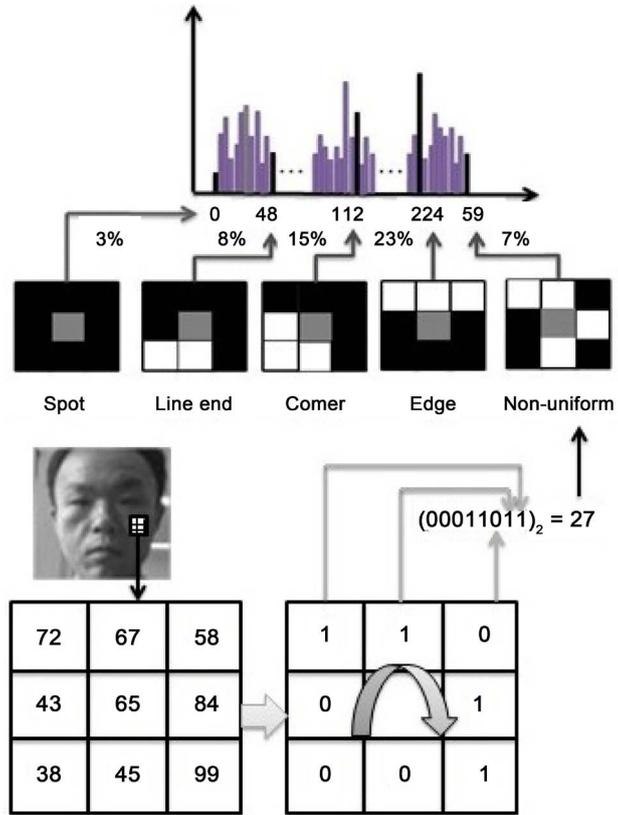


Figure 2.  $LBP_{(8,2)}^{m,2}$  descriptor encoding.

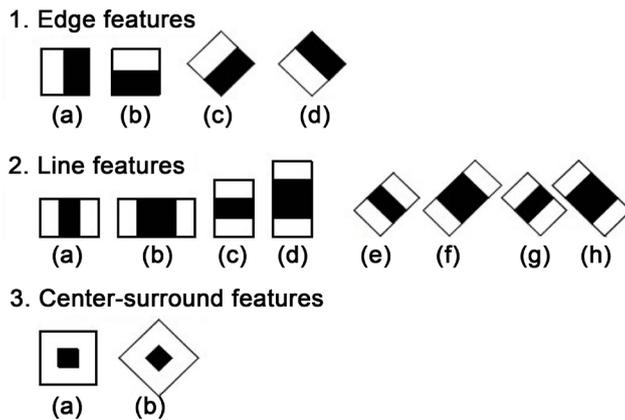


Figure 3. Haar-like features [9].

of the pixels in the black rectangular regions are subtracted from the values of the pixels in the white rectangular regions. The total represents the value of a Haar-like feature. While a Haar-like feature goes through the detection window, the area with the minimum value is the best match for this feature.

### 3. Face Detection Algorithms

#### 3.1. Face Detection Classifier

The Ada Boost algorithm [15] is used to extract the best features to detect the faces. The best features are chosen as weak classifiers and then concatenated to-

gether as a weighted combination of these features to construct a strong classifier, which is shown in the following equation:

$$S(x) = a_1 w_1 + a_2 w_2 + \dots + a_n w_n \quad (1)$$

In Equation (1),  $w_1(x), w_2(x), \dots, w_n(x)$  are  $n$  weak classifiers used to construct a strong classifier  $S(x)$ . The parameters  $a_1, a_2, \dots, a_n$  are weights associated with the  $n$  weak classifiers. The strong classifier can be used to detect faces with the following equation:

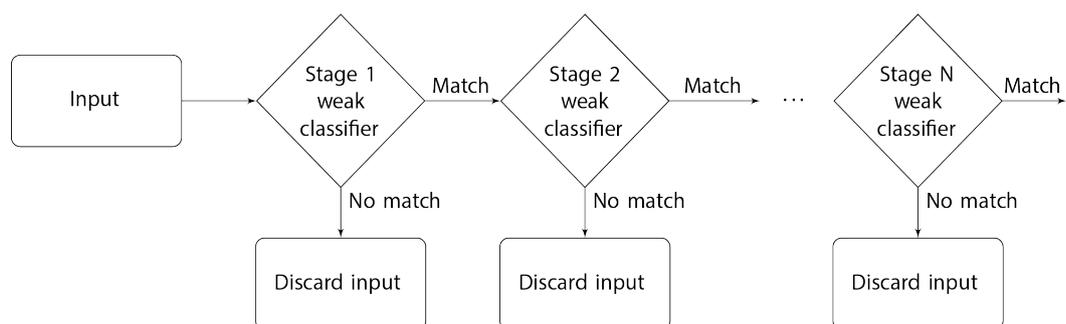
$$S_{th}(x) = \begin{cases} 1, & \text{if } S(x) \geq \frac{1}{2}(a_1 + a_2 + \dots + a_n) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

In Equation (2),  $S_{th}(x)$  is the threshold by the strong classifier to detect a face. "1" indicates that a face is present while "0" indicates that no face is detected. In our paper, the trained strong classifier  $S(x)$  can correctly detect faces with a high detection accuracy of 98.8%.

The cascade classifiers are trained using the Ada Boost algorithm. The cascade classifier consists in a series of tests on the input features, as shown in **Figure 4**. Selected features are separated into several stages and each stage is trained to be a strong classifier with best weak classifiers. The tested implementation uses 120 LBP and 32 Harr features for weak classifiers. Each stage is responsible for deciding whether the detection window might contain a face or not. The window will be discarded immediately once it fails at any stage. The result of this cascading is that the areas without faces will be discarded within the early stages, and therefore processed faster. The number of stages is defined during learning and is picked to achieve a predetermined detection accuracy.

The Chi-Squared difference [16] is used by the face detection classifier. The Chi-Squared difference is calculated between the LBP encoded histogram of a face detection region and the LBP encoded histogram of a predefined template image which is obtained by averaging 2400 facial images. The images used contain faces of various skin-colors, sexes, ages and are all picked from the MIT CBCL face database. Then, the difference is compared with a predefined threshold for classification. The Chi-Square difference equation is:

$$\chi^2(D, T) = \frac{1}{2} \sum_i \frac{(D_i + T_i)^2}{D_i + T_i} \quad (3)$$

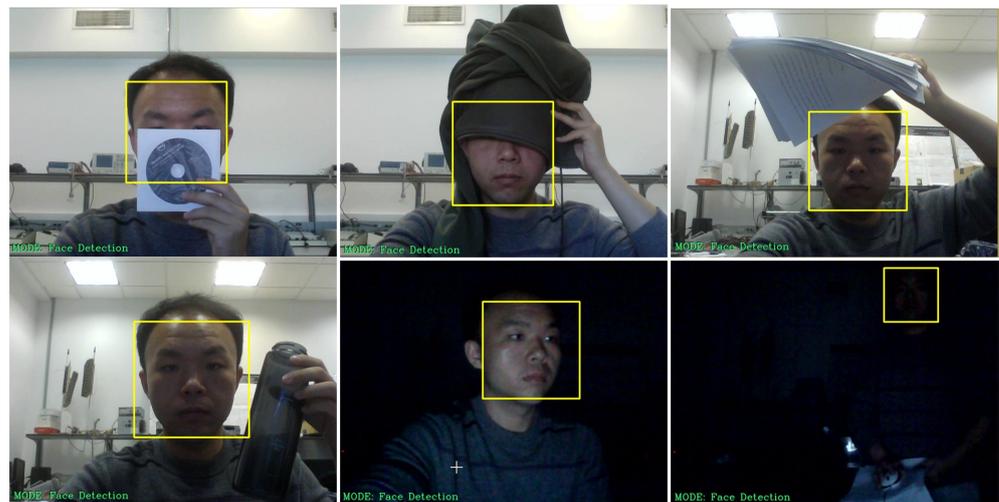


**Figure 4.** Flowchart of a cascading classifier.

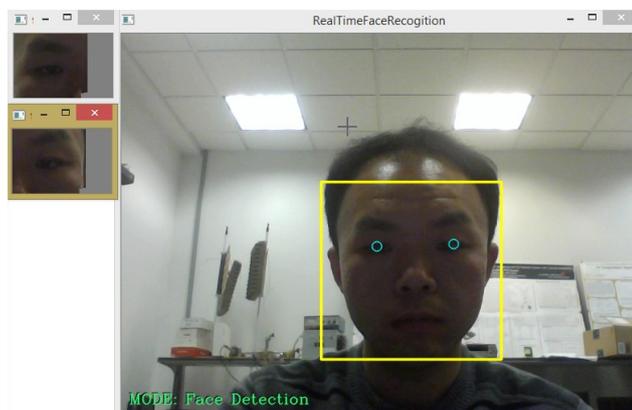
In Equation (3),  $D_i$  and  $T_i$  are the numbers of features in the  $i$ -th bin of the LBP encoded histogram of the detection region and the template image respectively. If the Chi-Squared difference is smaller than the threshold, it means that the detection window contains a face. Results of the face detection in various conditions are shown in **Figure 5**.

### 3.2. Eyes Detection

The Haar-like descriptor is used to detect both eyes of the face in order to enhance the face detection accuracy. The origin of the coordinate system for the facial image is chosen to be the top-left point. Two rectangular eye-search regions with the same size are extracted from each facial image at four predefined positions. For the left eye, the region extends on the x axis from 10% of the image width to 38%, and for the y axis from 15% of the image height to 40%. Since the right eye's search region is symmetric with respect to left-eye search region, the same proportions are used from the other side of the image. **Figure 6** shows the result of the eyes detection algorithm.



**Figure 5.** Face Detection under Various Conditions. Upper left, occlusion at bottom; upper middle, occlusion on top; upper right, face in shadow; lower left; object near face; lower middle, direct light on face; lower right, poor light condition.



**Figure 6.** Eyes detection in eye-search regions.

## 4. Facial Recognition

### 4.1. Affine Transformation

An affine transformation [17] is used to rectify the orientation and scale of the detected facial images to improve accuracy of recognition. An affine matrix is adopted to scale the detected facial image to the desired size, and rotate it so that the two eyes are horizontal.

$$A_v = \begin{pmatrix} s_x \cos \theta & -s_x \sin \theta & t_x s_x \cos \theta - t_x s_x \sin \theta \\ s_x \sin \theta & s_y \cos \theta & t_y s_x \sin \theta + t_y s_y \cos \theta \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

In Equation (4),  $A_v$  is the affine matrix and  $s_x, s_y$  are the scaling ratios in the  $x, y$  directions.  $t_x, t_y$  are the translation factors in  $x, y$  directions.  $\theta$  is the rotation angle of the image. The position of each pixel of the original facial image is multiplied by the affine matrix to constitute the corrected image, with a resolution of  $70 \times 70$  pixels.

**Figure 7** shows a facial image after correction. The two eyes are now horizontal and the image is resized to a standard dimension. The image is cropped to only show the facial features and discard the background.

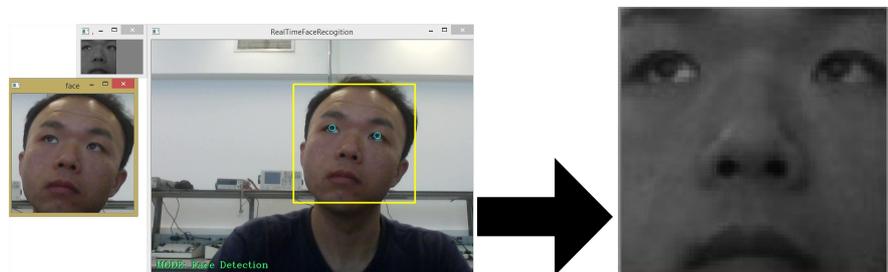
### 4.2. Histogram Equalization

The facial images of the same person can change drastically in various lighting conditions. A histogram equalization algorithm [18] is used to enhance the contrast of the detected facial images. The algorithm consists in replacing the pixel-values using a function designed to spread the repartition of the histogram. The function is given by the following Equation (5).

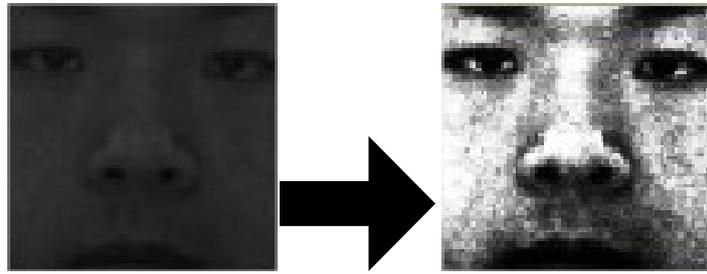
$$H(v) = \frac{CDF(v) - CDF_{\min}}{M \times N - CDF_{\min}} \times (L - 1) \quad (5)$$

In this equation,  $CDF(v)$  is the cumulative distribution function of pixels with value  $v$  for calculating the equalized value  $H(v)$ .  $M, N$  are the numbers of rows and columns for the facial image respectively.  $L$  is 256 and represents the gray-scale range.

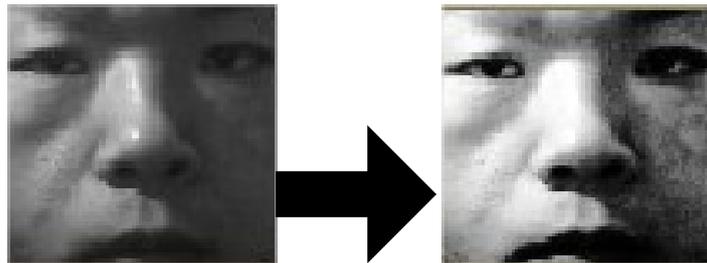
**Figure 8** shows the enhancement of the facial image using the histogram equalization algorithm. In strong lighting condition, however, one side of the face can be more exposed to the light than the other side, resulting in a significant lighting difference between the two sides. **Figure 9** shows an alternative



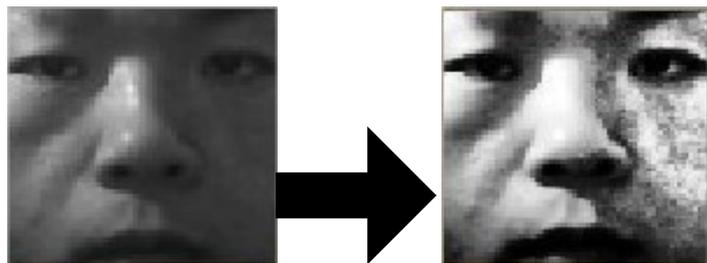
**Figure 7.** Affine transformation of a facial image.



**Figure 8.** Histogram equalization in weak lighting condition.



**Figure 9.** Separated histogram equalization in strong lighting condition.



**Figure 10.** Improved histogram equalization in strong lighting condition.

processing, applying the histogram equalization separately on both sides of the face.

In **Figure 9**, there is still a high lightning difference between both sides of the face, which might affect the recognition accuracy. In **Figure 10**, we propose an improved histogram equalization to decrease this lightning difference by mixing the separated histogram equalization with the whole-face histogram equalization gradually from the left or right edge to the center. Therefore, the far left or right region applies the separated histogram equalization and the central region smoothly mixes left or right equalized values and the whole-face equalized values.

### 4.3. Gaussian Filter

A Gaussian filter [19] is used to remove noise in the pre-processed facial images for a high facial recognition accuracy. A convolution matrix produced by a Gaussian function is used to smooth the facial images. The 2-D Gaussian function is given in Equation (6).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6)$$

The  $3 \times 3$  normalized convolution matrix with  $\theta = 0.84090$  is adopted for smoothing while preserving edges, which is shown in Equation (7).

$$H = \begin{bmatrix} 0.06163 & 0.12499 & 0.06163 \\ 0.12499 & 0.25350 & 0.12499 \\ 0.06163 & 0.12499 & 0.06163 \end{bmatrix} \quad (7)$$

The convolution process is defined by the Equation (8). For each pixel of the output image  $I'$ , the pixels of the original image  $I$  around this position are multiplied by the coefficients of the matrix  $H$ , and then summed up. The resulting image is a smaller image with a size of  $68 \times 68$ .

$$I'^{(i,j)} = \sum_{k=-1}^1 \sum_{t=-1}^1 I(i+k, j+t) \times H(k+2, t+2) \quad (8)$$

**Figure 11** shows that the Gaussian filter is removing the high-frequency noise in the pre-processed facial image.

#### 4.4. Principal Component Analysis

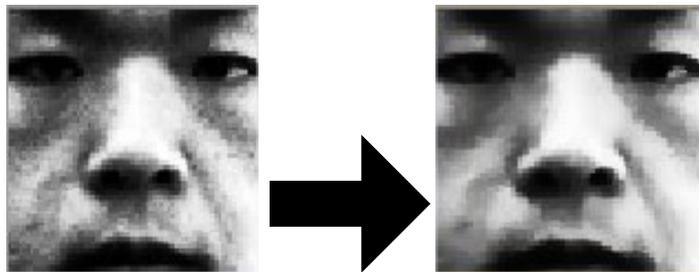
The desired facial images are first collected as samples for training the new coordinate system. Every pixel of the image is represented by a variable in one dimension for describing facial features, therefore the features of each desired facial image can be represented by a column vector with  $70 \times 70 = 4900$  dimensions. The PCA algorithm is used to recognize high-dimensional facial images with few principal components. The new base vectors  $\Phi' = (\Phi'_1, \Phi'_2, \dots, \Phi'_D)$  are given by maximizing the sample variance and minimizing the mean squared error.

$$\Phi' = \arg_{\Phi'} \min E \left[ (x - \hat{x})^2 \right] \quad (9)$$

$$\Phi'^T \Phi' = I \quad (10)$$

In Equation (9), the collected facial sample in the original coordinate system is represented as  $x$ . The collected facial sample which is reconstructed from the principal components is represented as  $\hat{x}$ . Equation (10) shows that each base vector is orthogonal to each other. The Lagrange multiplier is used to find the local minima of the function. The solution is shown in Equations ((11) and (12)).

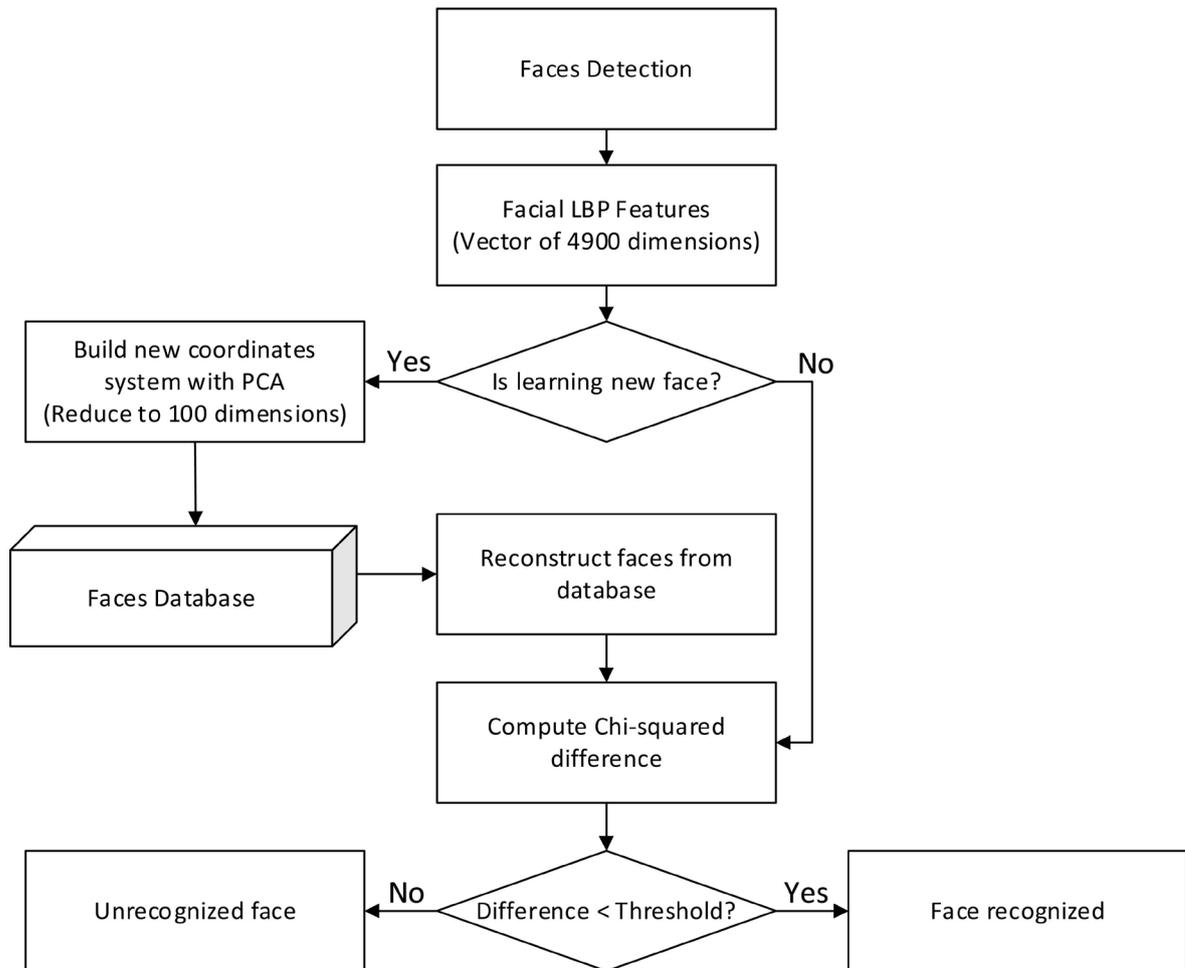
$$\sum_x \Phi'_i = \lambda'_i \Phi'_i \begin{cases} \lambda'_1 > \lambda'_2 > \dots > \lambda'_N \\ i = 1, 2, \dots, N \end{cases} \quad (11)$$



**Figure 11.** Gaussian smoothing.

$$\Sigma_x = E \left[ (x - \bar{x})(x - \bar{x})^T \right] = \frac{1}{P} \sum_{\mu=1}^P (x^{(\mu)} - \bar{x})(x^{(\mu)} - \bar{x})^T \quad (12)$$

$\Sigma_x$  is the covariance matrix of the sample vectors whose common features are removed by reducing the average vector of data.  $\bar{x}$  is the average vector.  $\lambda'_1, \lambda'_2, \dots, \lambda'_N$  are the eigenvalues of the covariance matrix.  $N$  indicates the dimension for each sample vector.  $P$  is the number of collected samples. The best base vector  $\Phi'_i$  is the eigenvector of the covariance matrix having the largest eigenvalue  $\lambda'_i$ . The flowchart of the PCA algorithm for facial recognition is shown in **Figure 12**. A value of  $D = 100$  is selected as the number of principal components to represent collected samples. A new face can be defined with only 100 dimensions, since the 100 principal components in the new coordinate system can illustrate most features of the new face. The projected values of each collected facial image on the 100 principal components are constructed into a 100-dimensional column vector for representing the training samples. If the difference between the reconstructed face and the new face is above the threshold of  $T = 0.4$ , it means that the new face was not recorded and it is displayed as an “unknown face”. Otherwise, the new face is identified as the sample face with the closest match.



**Figure 12.** Flowchart of facial recognition with PCA algorithm.

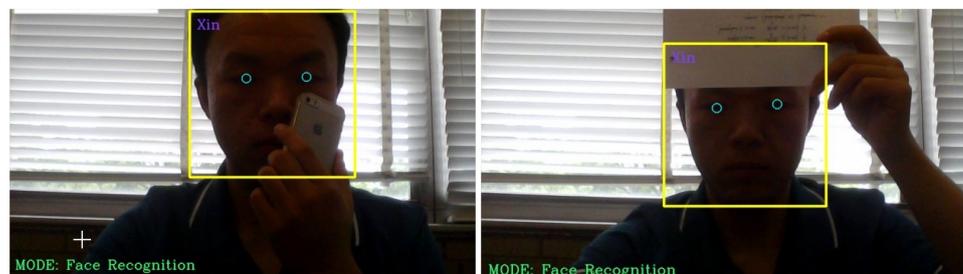
## 5. Results

The sample images used to train the face detector come from the MIT CBCL Face Database [20]. It includes 2492 faces with different identities, skin-colors, head poses, and 4548 non-faces images. The eye samples were extracted from the detected facial images in order to train the eye detector. In this paper, the algorithms were run on a computer with an Intel 2.50 GHz Core i7-3537U CPU at VGA resolution on a single thread. The processing time for detecting every face is of 11.4 ms and the processing time for detecting every pair of eyes within the facial regions is of 15.3 ms. With the help of the cascade classifier, the system is able to eliminate most non-facial features with little computational work. The resulting system is almost 3 times faster than the Joint Cascade detector [21] that takes 28.6 ms for face detection on a 2.93 GHz CPU at same resolution, and about 3000 times faster than Zhu *et al* detector [1] that detects every face in 33.8 s, still at VGA resolution. In order to test the face detection accuracy, 2836 faces and 3121 non-faces were randomly selected from the MIT CBCL Face Database [20] as well as the NIST Mugshot Identification Database [22] for cross validation purposes. By combining face normalization and eye detection, the algorithm achieves 98.8% detection accuracy and has a higher accuracy than other face detection algorithms, compared to 73.68% for the Color Based Segmentation [4], 97.14% for the Head Hunter [23]. It is to be noted that the fact that these two methods were tested on different databases but with similar properties. **Table 1** shows the test outcome for the facial detection, achieving a sensitivity of 99.2%, a specificity of 98.4% and a total accuracy of 98.8%. The Facial Recognition Technology Database [24], containing 3682 face samples of 526 subjects under various viewing conditions, is used to train the facial recognition algorithm and validate its results, resulting in a 99.2% positive recognition rate in this paper.

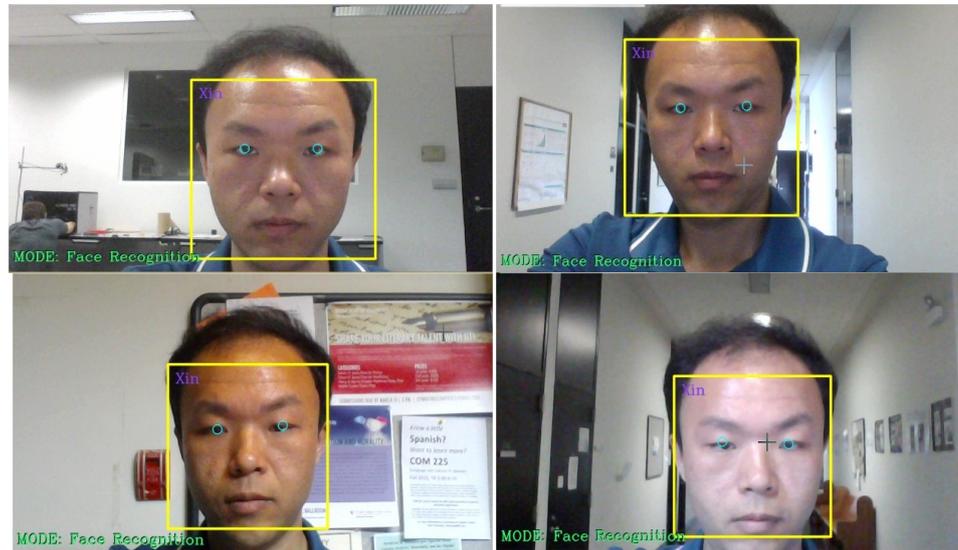
**Figure 13** shows that faces can be recognized in different real world conditions, such as picking up the cell phone or with occlusions on the hair. **Figure 14**

**Table 1.** Test outcome.

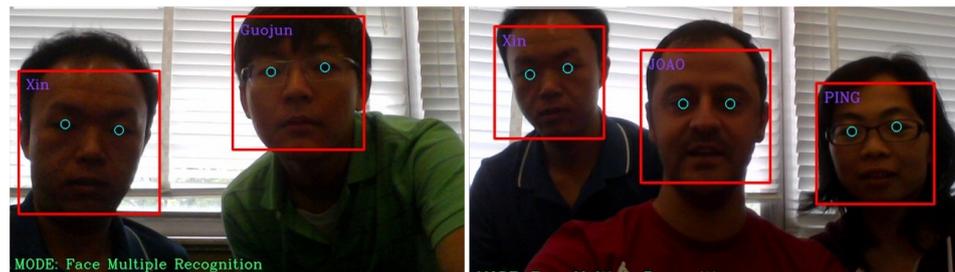
Input	Outcome	
	Face detected	No face detected
Face images (2836)	2813	39
Non-face images (2400)	23	2361



**Figure 13.** Real-time facial recognition in various conditions.



**Figure 14.** Real-time facial recognition in complex backgrounds.



**Figure 15.** Real-time multi-person facial recognition.

shows that faces can be recognized under various backgrounds. **Figure 15** shows that multiple faces can be recognized real-time.

## 6. Conclusions

Our algorithms can detect and recognize faces with high accuracy in real-time. It has a faster detection speed compared to other detection methods. The eyes detection is used to increase the face detection accuracy. The facial recognition performances are also greatly improved by using facial components alignment, contrast enhancement and image smoothing. Images of faces are collected as training samples in real-time and recognized under various conditions including among other faces.

Future work involves the training of new classifiers capable to expand the facial recognition to a wider range of facial orientations. The head rotation can be estimated so that the algorithm can correct further the facial image and maintain an accurate recognition.

## References

- [1] Zhu, X. and Ramanan, D. (2012) Face Detection, Pose Estimation and Landmark Localization in the Wild. *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, 16-21 June 2012, 2879-2886.

- [2] Tsitsoulis, A. and Bourbakis, N.G. (2015) A Methodology for Extracting Standing Human Bodies From Single Images. *IEEE Transactions on Human-Machine Systems*, **45**, 327-338. <https://doi.org/10.1109/THMS.2015.2398582>
- [3] Yanhun, Z. and Chongqing, L. (2003) Face Recognition Based on Support Vector Machine and Nearest Neighbor Classifier. *Journal of Systems Engineering and Electronics*, **14**, 73-76.
- [4] Tayal, Y., Lamba, R. and Padhee, S. (2012) Automatic Face Detection Using Color Based Segmentation. *International Journal of Scientific and Research Publications*, **2**, 1-7.
- [5] Tang, J., Deng, C., Huang, G.B. and Zhao, B. (2015) Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine. *IEEE Transactions on Geoscience and Remote Sensing*, **53**, 1174-1185. <https://doi.org/10.1109/TGRS.2014.2335751>
- [6] Su, C.Y. and Yang, J.F. (2014) Histogram of Gradient Phases: A New Local Descriptor for Face Recognition. *Computer Vision*, **8**, 556-567. <https://doi.org/10.1049/iet-cvi.2013.0208>
- [7] Pavithra, R., Usha Ruby, A. and Chellin Chandran, J.G. (2014) Scale Invariant Feature Transform Based Face Recognition from a Single Sample per Person. *International Journal of Computational Engineering Research*, **4**, 41-47.
- [8] Ahonen, T., Hadid, A. and Pietikainen, M. (2006) Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 2037-2041. <https://doi.org/10.1109/TPAMI.2006.244>
- [9] Lienhart, R. and Maydt, J. (2002) An Extended Set of Haar-Like Features for Rapid Object Detection. 2002 *International Conference on Image Processing*, Vol. 1, Rochester, 22-25 September 2002, I-900-I-903. <https://doi.org/10.1109/icip.2002.1038171>
- [10] Georgescu, D. (2011) A Real-Time Face Recognition System Using Eigenfaces. *Journal of Mobile, Embedded and Distributed Systems*, **3**, 193-204.
- [11] Li, Z., Lin, D. and Tang, X. (2009) Nonparametric Discriminant Analysis for Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, 755-761. <https://doi.org/10.1109/TPAMI.2008.174>
- [12] Ding, C., Xu, C. and Tao, D. (2015) Multi-Task Pose-Invariant Face Recognition. *IEEE Transactions on Image Processing*, **24**, 980-993. <https://doi.org/10.1109/TIP.2015.2390959>
- [13] Shen, C., Paisitkriangkrai, S. and Zhang, J. (2011) Efficiently Learning a Detection Cascade with Sparse Eigen-Vectors. *IEEE Transactions on Image Processing*, **20**, 22-35. <https://doi.org/10.1109/TIP.2010.2055880>
- [14] Maturana, D., Mery, D. and Soto, A. (2009) Face Recognition with Local Binary Patterns, Spatial Pyramid Histograms and Naive Bayes nearest Neighbor Classification. 2009 *International Conference of the Chilean Computer Science Society*, Santiago, 10-12 November 2009, 125-132. <https://doi.org/10.1109/SCCC.2009.21>
- [15] Mehmood, K. and Ahmad, B. (2013) Implementation of Face Detection System Using Adaptive Boosting Algorithm. *International Journal of Computer Applications*, **76**, 51-57. <https://doi.org/10.5120/13223-0639>
- [16] Noh, S. (2012)  $\chi^2$  Metric Learning for nearest Neighbor Classification and Its Analysis. 2012 *21st International Conference on Pattern Recognition*, Tsukuba, 11-15 November 2012, 991-995.
- [17] Pei, S.C. and Hsiao, Y.Z. (2015) Spatial Affine Transformations of Images by Using Fractional Shift Fourier Transform. 2015 *IEEE International Symposium on Circuits and Systems*, Lisbon, 24-27 May 2015, 1586-1589. <https://doi.org/10.1109/ISCAS.2015.7168951>
- [18] Peddigari, V.R., Srinivasa, P. and Kumar, R. (2015) Enhanced ICA Based Face Rec-

- ognition Using Histogram Equalization and Mirror Image Superposition. 2015 *IEEE International Conference on Consumer Electronics*, Las Vegas, 9-12 January 2015, 625-628. <https://doi.org/10.1109/ICCE.2015.7066555>
- [19] Reisert, M. and Burkhardt, H. (2008) Complex Derivative Filters. *IEEE Transactions on Image Processing*, **17**, 2265-2274. <https://doi.org/10.1109/TIP.2008.2006601>
- [20] Center for Biological and Computational Learning at MIT and MIT. CBCL Face Database. <http://cbcl.mit.edu/projects/cbcl/software-datasets/FaceData1Readme.html>
- [21] Chen, D., Ren, S., Wei, Y., Cao, X. and Sun, J. (2014) Joint Cascade Face Detection and Alignment. *Computer Vision ECCV*, Zurich, 6-12 September 2014, 109-122.
- [22] National Institute of Standards and Technology. NIST Mugshot Identification Database. <http://www.nist.gov/srd/nistsd18.cfm>
- [23] Mathias, M., Benenson, R., Pedersoli, M. and Van Gool, L. (2014) Face Detection without Bells and Whistles. *Computer Vision ECCV*, Zurich, 6-12 September 2014, Vol. 8692, 720-735.
- [24] FERET Program. The Facial Recognition Technology Database. <http://www.itl.nist.gov/iad/humanid/feret/>



**Submit or recommend next manuscript to SCIRP and we will provide best service for you:**

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.  
A wide selection of journals (inclusive of 9 subjects, more than 200 journals)  
Providing 24-hour high-quality service  
User-friendly online submission system  
Fair and swift peer-review system  
Efficient typesetting and proofreading procedure  
Display of the result of downloads and visits, as well as the number of cited articles  
Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact [jsip@scirp.org](mailto:jsip@scirp.org)