Traitement du Signal
Vol. 37, No. 6, December, 2020, pp. 929-937
Journal homepage: http://iieta.org/journals/ts

# A Face Detection Method Based on Skin Color Model and Improved AdaBoost Algorithm

Xiaoying Yang, Nannan Liang*, Wei Zhou, Hongmei Lu

School of Information Engineering, Suzhou University, Suzhou 234000, China

Corresponding Author Email: liangnannan@ahszu.edu.cn

https://doi.org/10.18280/ts.370606

**Received:** 19 July 2020
**Accepted:** 1 October 2020

***Keywords:***
*face detection, image processing, skin color model, AdaBoost algorithm*

## ABSTRACT

This paper integrates skin color model and improved AdaBoost into a face detection method for high-resolution images with complex backgrounds. Firstly, the skin color areas were detected in a multi-color space. Each image was subject to adaptive brightness compensation, and converted into the $YC_bC_r$ space, and a skin color model was established to solve face similarity. After eliminating the background interference by morphological method, the skin color areas were segmented to obtain the candidate face areas. Next, the inertia weight control factors and random search factor were introduced to optimize the global search ability of particle swarm optimization (PSO). The improved PSO was adopted to optimize the initial connection weights and output thresholds of the neural network. After that, a strong AdaBoost classifier was designed based on optimized weak BPNN classifiers, and the weight distribution strategy of AdaBoost was further improved. Finally, the improved AdaBoost was employed to detect the final face areas among the candidate areas. Simulation results show that our face detection method achieved high detection rate at a fast speed, and lowered false detection rate and missed detection rate.

## 1. INTRODUCTION

Face detection means to detect the face from the image and segment it from the background. Currently, face detection is mainly realized based on knowledge, features, appearance, and template matching [1-3]. The face cannot be detected correctly in a highly complex image, using one feature alone. As a result, the fusion of multiple features becomes the trend in face detection.

Faces can be detected simply, rapidly, and adaptively based on skin color. But this approach has a high false detection rate. It often mistakes areas with skin color or skin-like color for faces. Moreover, this approach is severely affected by light [4]. The AdaBoost-based face detection method boasts a high detection rate and a low false detection rate for images of frontal faces in simple backgrounds. However, this method faces several defects: the long training process, the proportional growth between detection rate and false detection rate, and the poor performance on images of side faces with complex backgrounds [5-7].

Many real-world scenarios, especially some embedded systems, require high resolution images with complex backgrounds, which calls for high detection speed and accuracy. However, the face often occupies a small portion in these high-resolution images. The detection speed can be accelerated by excluding the many non-face areas from the images. Therefore, this paper combines skin color model and improved AdaBoost into a novel face detection method.

Firstly, the skin color areas were detected in a multi-color space, and each color image was converted into the $YC_bC_r$ space. The brightness of the image was compensated for in an adaptive manner, and skin color model was constructed to solve the face similarity. After removing the background interference by morphological method, the skin color areas were segmented to exclude most non-face areas, leaving the face areas. Next, the AdaBoost training process was improved. The improved particle swarm optimization (PSO) was introduced to optimize the weights and thresholds of the neural network. Then, multiple new backpropagation neural network (BPNN) weak classifiers were merged into a strong AdaBoost classifier. Finally, the improved AdaBoost was applied to verify the face areas and non-face areas among the candidate areas, which greatly reduces the training time and improves the detection efficiency.

The main contributions of this paper are as follows:

(1) Light compensation was performed on each image. The skin color areas were detected in the multi-color space, and morphologically segmented to obtain the candidate face areas.

(2) Based on the PSO, two factors were added to control the change speed of the inertia weight, and the random search factor was introduced to control the degree of random influence of the current position and velocity on the next position.

(3) The improved PSO was applied to optimize the initial connection weights and output thresholds of the neural network, thereby avoiding the local minimum and enhancing network performance.

(4) Multiple improved BPNN weak classifiers were combined into a strong AdaBoost classifier with excellent performance; the weight distribution strategy of the AdaBoost was modified to assign greater weights to the correctly recognized positive samples: the increasing function of the sum of the weights of such samples in weak classifiers was added to the calculation of the predicted weights.

(5) The candidate face areas obtained by the skin color model were further screened by the improved AdaBoost, which reduces the detection time and improves the detection accuracy.

The remainder of this paper is organized as follows: Section 2 reviews the relevant works; Section 3 describes the skin color model; Section 4 introduces the proposed method that couples the skin color model with improved AdaBoost; Section 5 verifies the effectiveness of our method; Section 6 summarizes the research findings.

## 2. LITERATURE REVIEW

Face detection is an important application and a key research direction of digital image processing [8, 9]. The existing face detection methods are either based on features or grounded on statistical rules. The feature-based methods rely on knowledge rules, features, or templates. The knowledge rule-based strategies are generally inefficient [10]; the feature-based strategies cannot effectively capture some facial features, when the pose, angle, or expression changes; the template-based strategies involve complex calculations and difficulty in setting up a standard model [11, 12]. The statistical rule-based methods detect faces based on the common features learned from massive face samples. A typical example of statistical rule-based method is AdaBoost [6].

AdaBoost boasts good robustness, high detection rate, and low false detection rate in face detection. However, this algorithm does not apply to scenarios with high real-time requirements, because it needs to be trained for a long time on numerous samples. To solve the problem, AdaBoost has been studied and improved by many scholars. Zakaria and Suandi [13] combined neural network and AdaBoost into a face detection algorithm, which improves the detection performance by making BPNN the weak classifier of AdaBoost; But the algorithm is too complex to complete detection rapidly. Lee et al. [14] proposed a new weight adjustment factor, and applied it to a weighted support vector machine (SVM), serving as a weak learner of the AdaBoost. Li et al. [15] established a Gaussian model for skin color distribution, obtained the candidate areas containing face skin colors, and detected the skin color areas with a cascade classifier. Zakaria et al. [16] developed the hierarchical skin AdaBoost neural network (H-SKANN), in which the skin module roughly locates the faces, the AdaBoost filters out the non-face candidate areas, and the neural network acts as the main filter to finally detect faces.

In addition, Zhang and Ye [17] improved the AdaBoost based on dual features: the thresholds that correspond to the optimal values of the two features were searched for by the PSO, and the resulting dual feature weak classifiers were combined into a strong classifier. Han et al. [18] developed a fast pedestrian detection algorithm based on auto-encoding neural network and AdaBoost: the images were processed by the pedestrian detection algorithm based on aggregated channel feature (ACF) model; the subareas thus collected were normalized; the histogram of oriented gradient (HOG) features were extracted, and imported to the auto-encoding neural network; the AdaBoost was applied for class detection. Zhang and Fan [19] introduced the Q-statistic correlation determination into the training of weak classifiers to reduce the similarity between weak classifiers and remove similar rectangular features. To reduce the computing load of classic AdaBoost on massive training data, Taherkhania et al. [20] designed AdaBoost-convolutional neural network (CNN) to reduce the learning time required for component estimation. The above new methods either shorten the training time or improve the detection rate. None of them fully solves the defects of AdaBoost-based face detection.

Therefore, this paper creates a new face detection method based on skin color model and improved AdaBoost. Firstly, the skin color areas were detected in a multi-color space, and segmented by the skin color model and morphological method, producing the candidate face areas. To avoid local minimum, the weights and thresholds of the neural network were optimized by an improved PSO: the change speed of inertia weight is controlled by two factors, and the degree of random influence of the current position and velocity on the next position is controlled by a random search factor. On this basis, multiple weak BPNN classifiers were combined into a strong AdaBoost classifier. The weight distribution strategy of AdaBoost was improved to assign greater weights to the correctly recognized positive samples, thereby improving the detection rate. The improved AdaBoost was applied to verify the faces and non-faces among the candidate areas, and was proved to reduce training time, enhance classification performance, and achieve better detection accuracy.

## 3. SKIN COLOR MODEL

The skin color of faces differs greatly from image background. Face detection can be implemented accurately based on the invariance of skin colors to rotation and expressions. Studies have shown that brightness is the main contributor to the difference between skin colors. To separate the skin color areas, it is necessary to separate brightness from chroma in the face images. The key lies in the selection of a color space suitable for skin color segmentation.

### 3.1 $YC_bC_r$ color space

There are two important considerations in the selection between color spaces: the separability between color and brightness, and the clustering probability of skin colors. There are many mathematical models for color spaces, namely, RGB, HSV, and $YC_bC_r$. These color spaces are mutually convertible.

In the $YC_bC_r$ color space, the brightness is contained in the Y component, while the chroma is contained in $C_b$ and $C_r$. The two kinds of components need to be differentiated to process brightness and chroma separately. Both $C_b$ and $C_r$ obey two-dimensional (2D) independent distribution. The skin color points can be easily clustered, ensuring the segmentation effect. In addition, the $YC_bC_r$ color space can be converted from the RGB space at ease.

Thus, $YC_bC_r$ color space was adopted to build a Gaussian skin color model for face detection. First, the target image was mapped from the RGB color space to $YC_bC_r$ color space. Then, the Gaussian skin color model, which was built on skin color samples, was adopted to detect the skin colors, and output the detection results. The workflow of this model is illustrated in Figure 1.

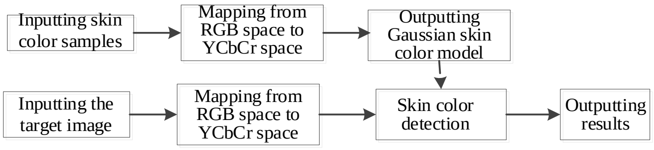

Inputting skin color samples
Mapping from RGB space to YCbCr space
Outputting Gaussian skin color model
Inputting the target image
Mapping from RGB space to YCbCr space
Skin color detection
Outputting results

**Figure 1.** The workflow of the Gaussian skin color model

*YC_bC_r* color space model [21] is one of the most important color systems for image segmentation, in which $Y$, $C_b$, and $C_r$ are the brightness component, blue chroma component, red chroma component, respectively. The $YC_bC_r$ can be linearly transformed into RGB by:

$$\begin{cases} Y = 0.257R + 0.564G + 0.098B + 16 \\ C_b = -0.148R - 0.291G + 0.439B + 128 \\ C_r = 0.439R - 0.368G - 0.071B + 128 \end{cases} \tag{1}$$

Through the transform, we have:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{255}\begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -137.797 & -74.494 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2}$$

The skin colors fall into the following ranges:

$$\begin{cases} 140 \le C_b \le 195 \\ 140 \le C_r \le 165 \end{cases} \tag{3}$$

The linear relationship with RGB blesses $YC_bC_r$ with a low complexity in color space and a good clustering property of skin colors. Therefore, it is not difficult to separate brightness and chroma in this color space.

### 3.2 Gaussian model

The skin colors are commonly simulated by the ellipse model, statistical histogram model, and Gaussian model [22-24]. The Gaussian model stands out for its high detection rate, fast calculation, and good representation of skin color distribution. Therefore, this paper models the skin color pixels in $YC_bC_r$ color space, using the simple Gaussian model. Once established, the Gaussian model was used to compute the similarity of pixels, and realize the segmentation of skin colors. The $YC_bC_r$ color space was extracted with Gaussian model, i.e., the $C_bC_r$ values in the selected areas were processed by:

$$\begin{cases} m = E(x) \\ x = (C_bC_r)^T \\ C = E\{(x-m)(x-m)^T\} \end{cases} \tag{4}$$

where, $m$ and $C$ are the mean and covariance matrix, respectively. The skin colors vary greatly in brightness, but insignificantly in chroma. Thus, the 2D Gaussian model works the same for the skin colors of different persons:

$$m = (\overline{\mathrm{C}_b}, \overline{\mathrm{C}_r}) \tag{5}$$

$$\overline{\mathrm{C}_b} = \frac{1}{N}\sum_{i=1}^{N} C_{b_i} \tag{6}$$

$$\overline{\mathrm{C}_r} = \frac{1}{N}\sum_{i=1}^{N} C_{r_i} \tag{7}$$

$$V = \begin{bmatrix} \sigma^2 C_r & \sigma_{C_r}\sigma_{C_b} \\ \sigma_{C_b}\sigma_{C_r} & \sigma^2 C_b \end{bmatrix} \tag{8}$$

where, $\overline{\mathrm{C}_b}$ and $\overline{\mathrm{C}_r}$ are the means of $C_b$, and $C_r$, respectively; $V$ is the covariance matrix. The means and covariance matrix were obtained by calculation based on experimental results. The $m$ and $C$ values fall into the following ranges:

$$\begin{cases} m = [117.4316 \;\; 148.5599] \\ C = \begin{bmatrix} 97.0946 & 24.4700 \\ 24.4700 & 141.9966 \end{bmatrix} \end{cases} \tag{9}$$

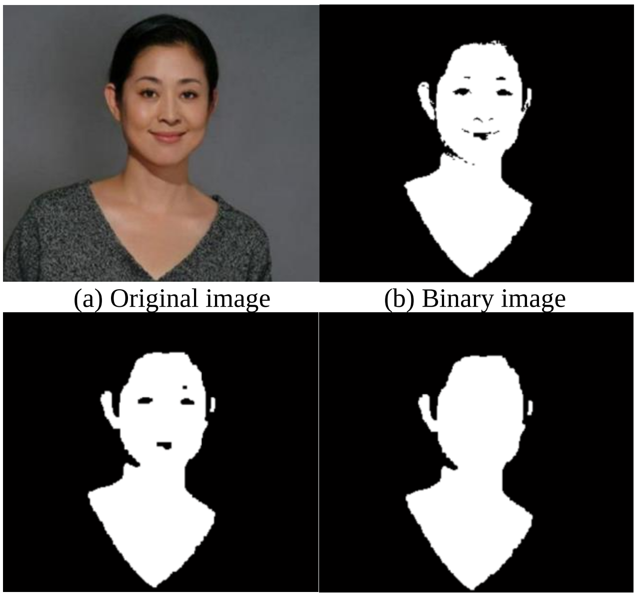
The established Gaussian model was used to judge whether a random pixel in the color image belongs to the skin. The probability of skin color judgement can be expressed as:

$$p(C_bC_r) = exp\,[-0.5(x-m)^T C^{-1}(X-m)] \tag{10}$$

This probability is positively correlated with the possibility for the pixel to be a skin pixel.

### 3.3 Skin color segmentation

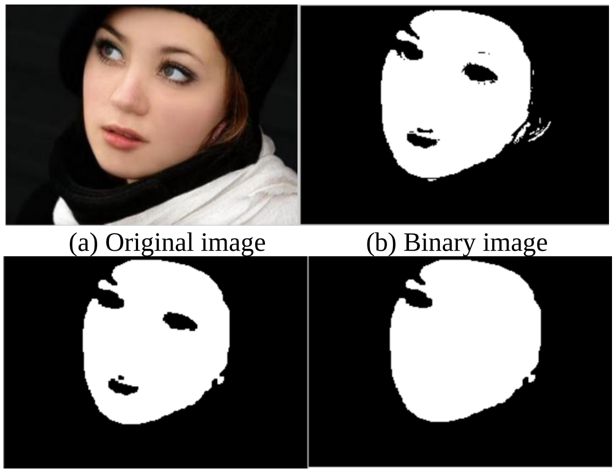
Since image brightness depends on light, the target RGB image should go through brightness compensation before skin color detection. After compensation, the image could be mapped from RGB color space to $YC_bC_r$ color space. Then, the face similarity map could be obtained by the Gaussian model. Based on experimental results, the threshold for binarization of the similarity map was optimized as 0.4.



(a) Original image (b) Binary image

(c) Morphologically processed image (d) Candidate face area

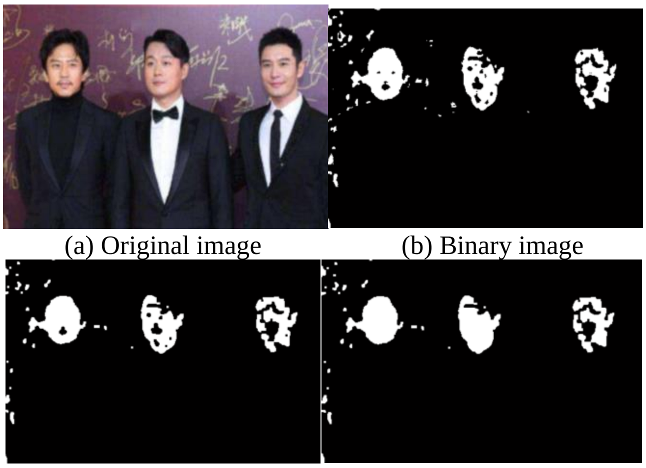
**Figure 2.** The processing results on a single front face



(a) Original image (b) Binary image

(c) Morphologically processed image (d) Candidate face area

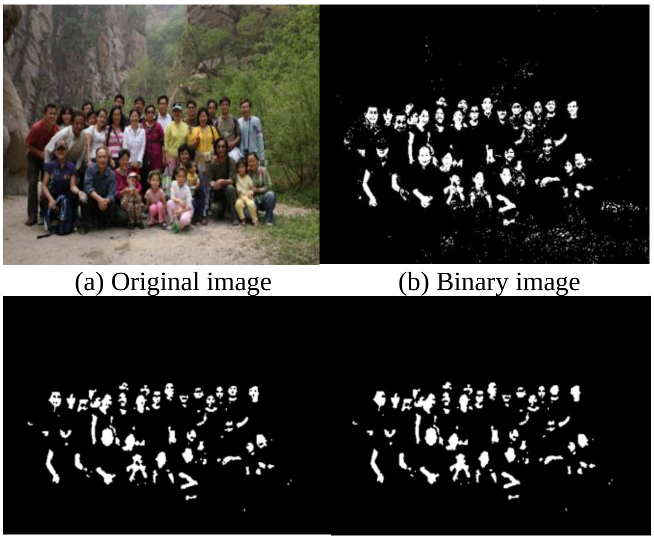
**Figure 3.** The processing results on a single side face



(a) Original image (b) Binary image

(c) Morphologically processed image (d) Candidate face area

**Figure 4.** The processing results on multiple faces



(a) Original image (b) Binary image

(c) Morphologically processed image (d) Candidate face area

**Figure 5.** The processing results on multiple faces with complex background

In the binary image, some small areas might be misjudged as skin colors, due to the background and noise arising in image collection or processing. To eliminate the scattered outliers in the face image, morphological operation was performed to erode, expand, and smoothen the image, and fill up the holes in skin color areas, resulting in the candidate face areas. Figures 2-5 present the processing results on a single front face, a single side face, multiple faces, and multiple faces with complex background, respectively.

## 4. FACE DETECTION METHOD

### 4.1 Traditional BPNN

Capable of solving nonlinear complex system problems, the BPNN is a multi-layer feedforward neural network with one-way propagation [25]. As shown in Figure 6, a typical BPNN consists of an input layer, a hidden layer, and an output layer.

After the training samples were obtained, the activation value of input layer nodes was inputted to the input layer, which was propagated to the output layer via the hidden layer. Then, the output layer nodes produced the network output. Through the training process, the connection weights of the network were modified to reduce the error between actual and expected outputs, making the result more desirable. The BPNN performance is measured by mean squared error $M_{SE}$:

$$\mathrm{M}_{SE}=\frac{1}{2}\sum_{i=1}^{m}(x_i-y_i)^2 \qquad (11)$$

where, $x_i$ and $y_i$ are the expected and actual outputs of the network, respectively; $i$ and $m$ are the serial number of nodes, and the number of output layer nodes, respectively. The learning process intends to reduce the value of formula (11) to an acceptable level.

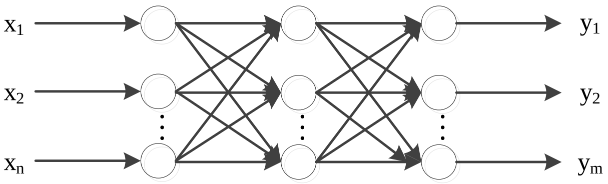


**Figure 6.** The structure of the BPNN with one hidden layer

### 4.2 BPNN optimized by improved PSO

Mimicking biological mechanisms, the PSO converges to the global optimal solution by iteratively searching for the best-known solutions [26]. Traditional PSO updates the position and velocity of each particles by:

$$\vartheta_{id}(t+1)=\omega\vartheta_{id}(t)+c_1r_{1d}\big(p_{id}-x_{id}(t)\big)+c_2r_{2d}\Big(p_{gd}-x_{id}(t)\Big) \qquad (12)$$

$$(t+1)=x_{id}(t)+\vartheta_{id}(t+1) \qquad (13)$$

$$\omega(t)=(\omega_{max}-\omega_{min}-d_1)exp\left(\frac{1}{1+\frac{d_2t}{t_{max}}}\right) \qquad (14)$$

where, $\vartheta_{id}(t)$, and $x_{id}(t)$ are the velocity and position of particle $i$ at time $t$, respectively; $c_1$ and $c_2$ are the learning factors that adjusts the step lengths to ensure the particle to move towards the best-known individual position $p_{id}$ and the best-known global position $p_{gd}$, respectively; $r_{1d}$ and $r_{2d}$ are random numbers in [0, 1]; $\omega$ is the inertia weight factor that controls the influence of previous velocity on the current velocity (a small $\omega$ favors local search, and a large $\omega$ favors global search.

The strategy of non-linear decreasing inertia weight was introduced to avoid local minimum. Specifically, the control factors $d_1$ and $d_2$ were added to regulate the change speed of the inertia weight between $\omega_{max}$ and $\omega_{min}$. As shown in formula (14), the control factors enhance the global and local search abilities, enabling the algorithm to converge faster to global optimal solution. The maximum number of iterations $t_{max}$ was set to 100. The other parameters were configured as $\omega_{max}$=0.9, $\omega_{min}$=0.4, $d_1$=0.2, and $d_2$=0.7.

Next, the random search factor γ was added to represent the degree of random influence of the current position and velocity on the next position:

$$\gamma = \frac{X_{id}(t+1)}{X_{id}(t) + v_{id}(t+1)} \tag{15}$$

To verify its effectiveness, the improved PSO was compared with the traditional PSO. The results in Figure 7 show that the improved PSO realized better convergence speed and accuracy than the traditional PSO. Hence, the improved PSO was applied to optimize the initial weights and thresholds of the BPNN.

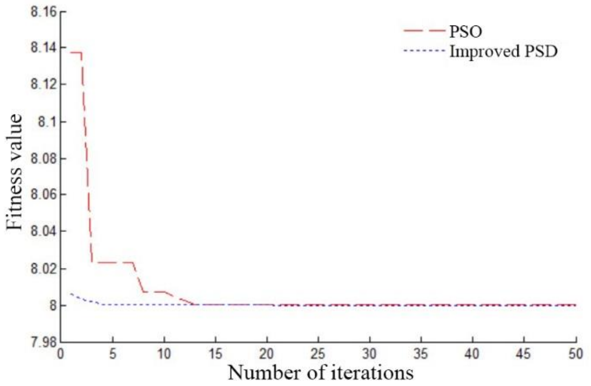


**Figure 7.** The fitness values of improved PSO and traditional PSO

In the BPNN, the weights and thresholds are generally initialized as random numbers in [-0.5, 0.5]. The initial parameters have major impact on the network performance. But reasonable parameter values are not easy to obtain. Here, the initial weights and thresholds were optimized by the improved PSO through population initialization, fitness calculation, and update of velocity and position.

Since the BPNN structure is already known, the number of weights and thresholds could be directly determined. All the weights and thresholds are contained in each particle of the population. The fitness of each particle can be obtained by the fitness function. Then, the particle of the optimal fitness in each iteration could be found through velocity and position updates. On this basis, the weights and thresholds of the network can be optimized. The PSO optimization of BPNN covers the following steps:

Step 1. Initialize the parameters of the improved PSO. Since all the weights and thresholds of the BPNN are contained in particles, the weights and thresholds can be determined based on the network structure.

Step 2. Determine the initial weights and thresholds of the BPNN based on the particles. The BPNN was trained by the training samples, and adopted for prediction. The fitness of each particle is defined as the absolute error between the predicted output and expected output:

$$fit = q(\sum_{i=1}^{n} |eo_i - o_i|) \tag{16}$$

where, $q$ is a coefficient; n is the number of output layer nodes; $eo_i$, and $o_i$ are the predicted output and expected output of node i in the BPNN, respectively.

Step 3. Compare the current fitness of each particle with its best-known fitness of position. If the current fitness is better, save it as the best-known fitness of position.

Step 4. Update the position and velocity of each particle by formulas (12) and (13).

Step 4. If the termination condition is satisfied, output the optimal individual, and assign the optimal weights and thresholds to the BPNN; otherwise, return to Step 2.

### 4.3 Improved AdaBoost

The core of AdaBoost is to minimize the error between expected output and actual output by combining multiple weak classifiers. To improve classification performance, the optimized BPNN was taken as a novel weak classifier, and used to build a strong classifier. The relevant procedure is as follows:

Step 1. Collect data and initialize the network. Randomly extract M groups of data as the training samples, initialize their weights by formula (17):

$$D_t(i) = \frac{1}{M} \quad (i \in 1, 2, \cdots, M) \tag{17}$$

The network structure depends on the input and output dimensions of the samples. The initial weights and thresholds were optimized by the improved PSO.

Step 2. Set up weak classifiers. Train the BPNN optimized by improved PSO with the training samples to obtain the sum of predicted errors of the t-th weak BPNN classifier:

$$e_t = \sum [g_t(i) - y_i] \quad (i \in 1, 2, \cdots, M) \tag{18}$$

where, $g_t(i)$ and $y_i$ are the expected and actual outputs of the network, respectively.

Step 3. Calculate predicted weights. The weight of each weak BPNN classifier depends on the predicted error of the training samples and $e_t$. For multi-class problems, the traditional AdaBoost finds multiple weak classifiers whose error rate is smaller than 1/2, by lowering the weights of correctly classified samples and increasing those of the samples incorrectly classified by the previous weak classifier. However, the error rate of weak classifiers tends to increase with the number of weak classifiers. As a result, the strong classifier made up of weak classifiers is rarely sufficiently good. Thus, the weight distribution strategy of AdaBoost was improved as:

$$a_t = \frac{1}{2} ln\left(\frac{1 - e_t}{e_t}\right) + k \times e^{p_t} \tag{19}$$

where, $p_t$ is the recognition ability of a weak classifier of positive samples, that is, the sum of the weights of all positive samples correctly recognized by the weak classifier in the t-th iteration. The new weighting parameters were solved by $k \times e^{p_t}$. Under the same error rate, the weak classifier good at recognizing positive samples was given a large weight.

Step 4. Adjust the weights. The weights of the training samples in the next round can be adjusted by $a_t$:

$$D_{t+1}(i) = \frac{D_t(i)}{B_t} \exp\left[-a_t g_t(i) y_i\right] \tag{20}$$

where, $B_t$ is the normalization factor; $D_t(i)$ is the weight adjusted through $t$-1 training cycles.

Step 5. Build the strong classifier. After T iterations, the T groups of weak classifiers can be combined into the strong classifier:

$$h(x) = sign(\sum_{t=1}^{T} a_t f_t(x)) \quad (21)$$

where, $f_t(x)$ is the t-th weak classifier. The workflow of the improved AdaBoost is shown in Figure 8.

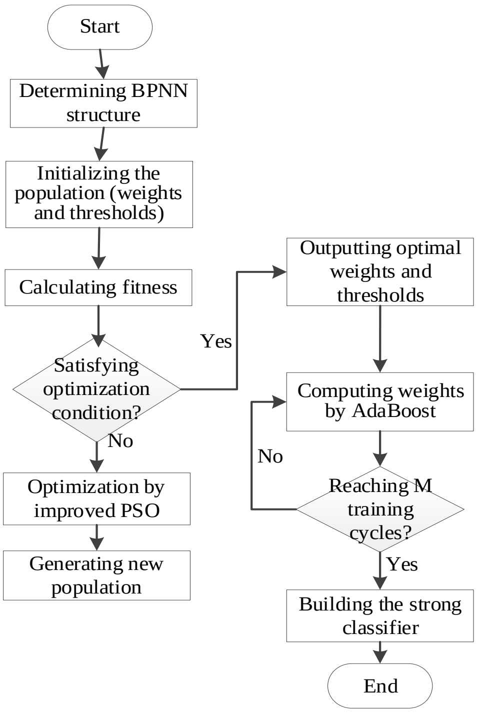

Start
Determining BPNN structure
Initializing the population (weights and thresholds)
Calculating fitness
Satisfying optimization condition?
Yes
No
Optimization by improved PSO
Generating new population
Outputting optimal weights and thresholds
Computing weights by AdaBoost
No
Reaching M training cycles?
Yes
Building the strong classifier
End

**Figure 8.** The workflow of improved AdaBoost

### 4.4 Integration between skin color model and improved AdaBoost

The skin color model operates at a fast speed. Under the effect of light, however, the model might mistake non-face areas or skin-like areas as faces, resulting in a high false alarm rate. The previous analysis shows that the improved AdaBoost is more reliable in face detection than the traditional AdaBoost. But the cascaded strong classifier in the improved AdaBoost works slowly, due to the scanning of the entire image. To sum up, skin color model and improved AdaBoost have their respective merits and defects in face detection.

Therefore, this paper fuses skin color model and improved AdaBoost into an innovative face detection method, capable of detecting faces accurately and rapidly. The main idea of the new method is as follows: The skin areas and skin-like areas (background) are segmented from the original image by the skin color model; the noises are removed by morphological operation; the denoised image is detected by the cascaded classifier of improved AdaBoost to obtain the final face areas. The workflow of the new method is explained in Figure 9.

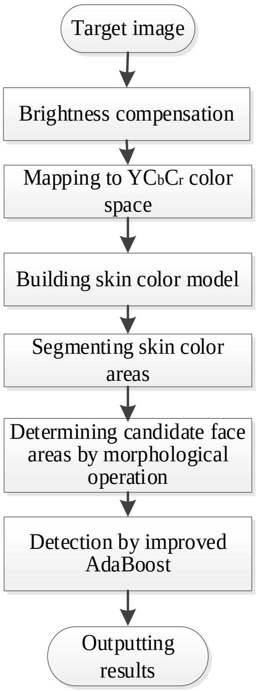

Target image
Brightness compensation
Mapping to YCbCr color space
Building skin color model
Segmenting skin color areas
Determining candidate face areas by morphological operation
Detection by improved AdaBoost
Outputting results

**Figure 9.** The workflow of the new face detection method

## 5. EXPERIMENTS AND RESULTS ANALYSIS

The proposed method was verified through simulations on MATLAB. The classifier of the improved AdaBoost was trained on the samples of MIT face database. A total of 500 images from different sources were selected to compare the performance of improved AdaBoost against that of traditional AdaBoost, including 350 single face images, and 150 multi-face images. There are in total 654 faces in the multi-face images. The accuracy and false detection rate of the two algorithms were calculated by:

$$\text{Accuracy} = \frac{\text{Number of correct detections}}{\text{Total number of faces}} \quad (22)$$

$$\text{False detection rate} = \frac{\text{Number of false detections}}{\text{Sum of false detections}} \quad (23)$$

**Table 1.** The face detection performance of the two algorithms

| Image type | Number of faces | Algorithm | Number of correct detections | Number of false detections | Number of missed detections | Detection rate | False detection rate | Missed detection rate |
|---|---|---|---|---|---|---|---|---|
| Single face | 350 | AdaBoost | 312 | 27 | 11 | 89.1% | 7.7% | 3.2% |
| | | Improved AdaBoost | 334 | 6 | 10 | 95.4% | 1.7% | 2.9% |
| Multi-face | 654 | AdaBoost | 548 | 56 | 50 | 83.8% | 8.6% | 7.6% |
| | | Improved AdaBoost | 615 | 11 | 28 | 94.1% | 1.6% | 4.3% |

As shown in Table 1, under the same simulation conditions, the improved AdaBoost achieved high detection rates, low false detection rates, and relatively low missed detection rates on both single face images and multi-face images. The good performance comes from the fusion of skin color model, and the optimization by the improved BPNN.

Further, precision and recall were adopted to evaluate the face detection performance of the two algorithms (Figures 10 and 11). The precision is positively correlated with detection accuracy, while recall is positively correlated with the comprehensiveness of detection:

$$\text{Precision} = \frac{\text{Number of correct detections}}{\text{Number of samples detected as faces}} \tag{24}$$

$$\text{Recall} = \frac{\text{Number of correct detections}}{\text{Number of all faces}} \tag{25}$$

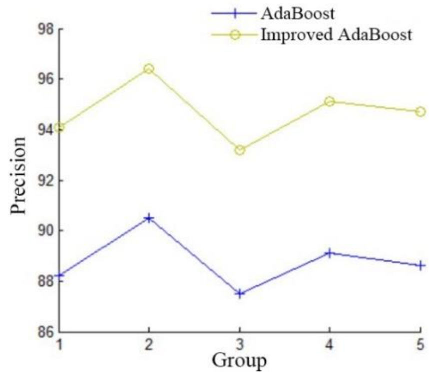


**Figure 10.** The precisions of the two algorithms

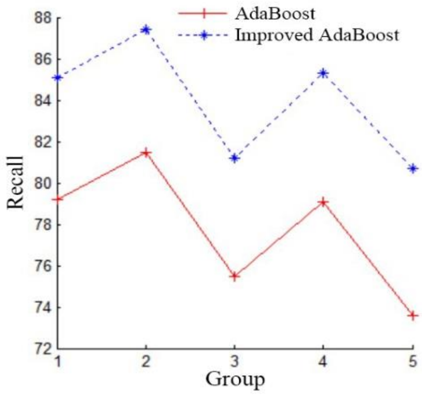
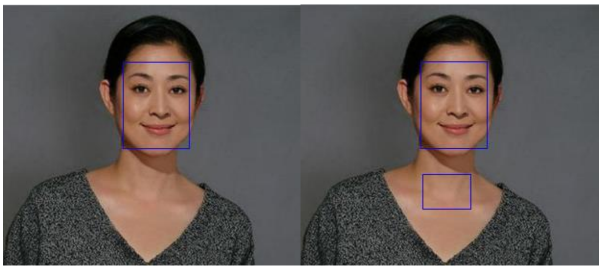


**Figure 11.** The recalls of the two algorithms

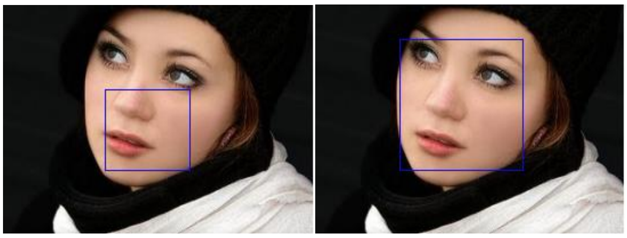
As shown in Figures 10 and 11, the improved algorithm had much higher precision and recall than the traditional AdaBoost.

Finally, the proposed method, which couples skin color model with improved AdaBoost, was compared with the traditional AdaBoost on face images in four scenarios: single front face image; single side face image; multi-face image; and multi-face image with complex background. The images were collected randomly from the Internet. The relevant results are shown in Figures 12-15 in turn.



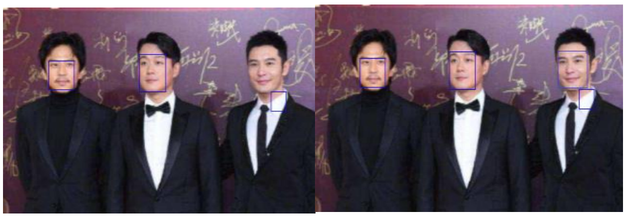
(a) AdaBoost (b) Our method

**Figure 12.** The detection results on a single front face



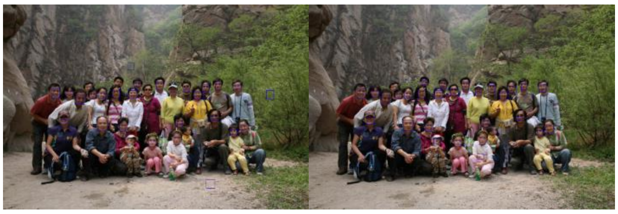
(a) AdaBoost (b) Our method

**Figure 13.** The detection results on a single side face



(a) AdaBoost (b) Our method

**Figure 14.** The detection results on multiple faces



(a) AdaBoost (b) Our method

**Figure 15.** The detection results on multiple faces with complex background

As shown in Figures 12-15, AdaBoost achieved relatively high detection rate. However, the images output by the algorithm contained non-face areas (background) that were misjudged as faces. In particular, the algorithm had high missed and false detection rates in the multi-face image with complex background.

Our method fuses skin color model with improved AdaBoost. Most face areas and non-face areas in the images were detected through the segmentation by skin color model. Even in the image with complex background, the two kinds of areas were detected effectively. The optimization of the improved BPNN greatly enhances the classification ability of the AdaBoost. Thus, the improved AdaBoost could eliminate the face areas that were incorrectly detected or missed by the skin color model. In addition, the improved AdaBoost only need to detect the images screened by the skin color model. For these reasons, the proposed method could achieve a high detection rate, and a low missed/false detection rate, at a fast speed.

## 6. CONCLUSIONS

This paper proposes a novel face detection method that integrates skin color model with improved AdaBoost. The initial face detection was realized by the skin color model: the target image was segmented by skin color, and the noise impact was eliminated through morphological operation, creating the candidate face areas. These areas were imported to the AdaBoost classifier instead of the entire image. The reduction in detection area greatly accelerates the detection speed. Moreover, the global and local search abilities of the PSO were improved by additional control factors and a random search factor. The improved PSO was used to optimize the connection weights and thresholds of the BPNN. After that, each optimized BPNN was treated as a weak classifier, and multiple weak BPNN classifiers were cascaded into a strong AdaBoost classifier. These moves elevate the detection rate, lower the false/missed detection rate, and shorten the detection time for face images. Considering the complexity of BPNN, the future research will further shorten the runtime of our method, and expedite the training speed, without scarifying its test accuracy.

## ACKNOWLEDGMENT

This work was supported by the third batch of reserve candidates for academic and technical leaders (Grant No.: 2018XJHB07), Teaching research project of Suzhou University (Grant No.: szxy2020jyxm06), Key curriculum construction project (Grant No.: szxy2018zdkc19), Large scale online open course (MOOC) demonstration project (Grant No.: 2019mooc300 and 2019mooc318), Professional leader of Suzhou University (Grant No.: 2019XJZY22), Anhui province's key R&D projects include Dabie Mountain and other old revolutionary base areas, Northern Anhui and poverty-stricken counties in 2019 (Grant No.: 201904f06020051), Key scientific research project of Suzhou University (Grant No.: 2019yzd04), research on multisource heterogeneous data acquisition, storage and intelligent analysis technology based on power big data platform (Grant No.: Z181100005118016).

## REFERENCES

[1] Deng, J., Trigeorgis, G., Zhou, Y., Zafeiriou, S. (2019). Joint multi-view face alignment in the wild. IEEE Transactions on Image Processing, 28(7): 3636-3648. https://doi.org/10.1109/TIP.2019.2899267

[2] Tang, F., Wu, X., Zhu, Z., Wan, Z., Chang, Y., Du, Z., Gu, L. (2020). An end-to-end face recognition method with alignment learning. Optik, 205: 164238. https://doi.org/10.1016/j.ijleo.2020.164238

[3] Zhou, M., Lin, H., Young, S.S., Yu, J. (2018). Hybrid sensing face detection and registration for low-light and unconstrained conditions. Applied Optics, 57(1): 69-78. https://doi.org/10.1364/AO.57.000069

[4] Ly, B.C.K., Dyer, E.B., Feig, J.L., Chien, A.L., Del Bino, S. (2020). Research techniques made simple: cutaneous colorimetry: A reliable technique for objective skin color measurement. Journal of Investigative Dermatology, 140(1): 3-12. https://doi.org/10.1016/j.jid.2019.11.003

[5] Rajeshwari, J., Karibasappa, K., Gopalkrishna, M.T. (2016). Adaboost modular tensor locality preservative projection: Face detection in video using Adaboost modular-based tensor locality preservative projections. IET Computer Vision, 10(7): 670-678. https://doi.org/10.1049/iet-cvi.2015.0406

[6] Du, S., Liu, J., Liu, Y., Zhang, X., Xue, J. (2017). Precise glasses detection algorithm for face with in-plane rotation. Multimedia Systems, 23(3): 293-302. https://doi.org/10.1007/s00530-015-0483-4

[7] Rajeshwari, J., Karibasappa, K., Gopalkrishna, M.T. (2016). Adaboost modular tensor locality preservative projection: face detection in video using Adaboost modular-based tensor locality preservative projections. IET Computer Vision, 10(7): 670-678. https://doi.org/10.1007/s00530-015-0483-4

[8] Bong, K., Choi, S., Kim, C., Yoo, H.J. (2017). Low-power convolutional neural network processor for a face-recognition system. IEEE Micro, 37(6): 30-38. https://doi.org/10.1109/MM.2017.4241350

[9] Yu, B., Tao, D. (2018). Anchor cascade for efficient face detection. IEEE Transactions on Image Processing, 28(5): 2490-2501. https://doi.org/10.1109/TIP.2018.2886790

[10] Lee, Y.B., Lee, S. (2011). Robust face detection based on knowledge-directed specification of bottom-up saliency. Etri Journal, 33(4): 600-610. https://doi.org/10.4218/etrij.11.1510.0123

[11] George, M., Sivan, A., Jose, B.R., Mathew, J. (2019). Real-time single-view face detection and face recognition based on aggregate channel feature. International Journal of Biometrics, 11(3): 207-221. https://doi.org/10.1504/IJBM.2019.100829

[12] Li, H., Hua, G., Lin, Z., Brandt, J., Yang, J. (2013). Probabilistic elastic part model for unsupervised face detector adaptation. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, pp. 793-800. https://doi.org/10.1109/ICCV.2013.103

[13] Zakaria, Z., Suandi, S.A. (2011). Face detection using combination of Neural Network and Adaboost. In TENCON 2011-2011 IEEE Region 10 Conference, Bali, Indonesia, pp. 335-338. https://doi.org/10.1109/TENCON.2011.6129120

[14] Lee, W., Jun, C.H., Lee, J.S. (2017). Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. Information Sciences, 381: 92-103. https://doi.org/10.1016/j.ins.2016.11.014

[15] Li, P., Wang, H., Li, Y., Liu, M. (2020). Analysis of face detection based on skin color characteristic and AdaBoost algorithm. In Journal of Physics: Conference Series, 1601(5): 052019. https://doi.org/10.1088/1742-6596/1601/5/052019

[16] Zakaria, Z., Suandi, S.A., Mohamad-Saleh, J. (2018). Hierarchical skin-adaboost-neural network (H-SKANN) for multi-face detection. Applied Soft Computing, 68: 172-190. https://doi.org/10.1016/j.asoc.2018.03.030

[17] Zhang, J., Ye, Q.W. (2020). Improved AdaBoost face detection algorithm based on dual features. Wireless Communication Technology, 29(2): 23-27.

[18] Han, X.Z., Li, D.F., Wang, K.J., Zhou, L.Y. (2018). Fast pedestrian detection algorithm based on auto-encoder neural network and adaboost. Journal of South-Central University for Nationalities (Natural Science Edition), 108-113.

[19] Zhang, J.C., Fan, W. (2011). AdaBoost face detection algorithm based on correlation. Computer Engineering, 37(8): 158-163. https://doi.org/10.3969/j.issn.1000-3428.2011.08.054

[20] Taherkhani, A., Cosma, G., McGinnity, T.M. (2020). AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. Neurocomputing, 404: 351-366.

https://doi.org/10.1016/j.neucom.2020.03.064

[21] Wicaksono, B.A., Novamizanti, L., Ibrahim, N. (2019). Tea leaf maturity levels based on ycbcr color space and clustering centroid. In Journal of Physics: Conference Series, 1367(1): 012028. https://doi.org/10.1088/1742-6596/1367/1/012028

[22] Kang, S., Choi, B., Jo, D. (2016). Faces detection method based on skin color modeling. Journal of Systems Architecture, 64: 100-109. https://doi.org/10.1016/j.sysarc.2015.11.009

[23] Veredas, F.J., Mesa, H., Morente, L. (2015). Efficient detection of wound-bed and peripheral skin with statistical colour models. Medical & Biological Engineering & Computing, 53(4): 345-359. https://doi.org/10.1007/s11517-014-1240-0

[24] Zhao, X., Li, Y., Zhao, Q. (2018). A fuzzy clustering approach for complex color image segmentation based on gaussian model with interactions between color planes and mixture gaussian model. International Journal of Fuzzy Systems, 20(1): 309-317. https://doi.org/10.1007/s40815-017-0411-1

[25] He, F., Zhang, L. (2018). Prediction model of end-point phosphorus content in BOF steelmaking process based on PCA and BP neural network. Journal of Process Control, 66: 51-58. https://doi.org/10.1016/j.jprocont.2018.03.005

[26] Yu, H., Tan, Y., Zeng, J., Sun, C., Jin, Y. (2018). Surrogate-assisted hierarchical particle swarm optimization. Information Sciences, 454: 59-72. https://doi.org/10.1016/j.ins.2018.04.062