Face Detection Based on the Improved AdaBoost Algorithm

Rui LI, Yingying LI*

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Abstract

In order to solve the overfitting of sample weights and the low detection rate in training process of the traditional AdaBoost algorithm, an improved AdaBoost algorithm based on Haar-like features and LBP features is proposed. This method improves weight updating rule and weights normalization rule of the traditional AdaBoost algorithm. Then combining this method with the AdaBoost algorithm based on Haar-like features and MB-LBP features fusion. The experiment result shows that this method can effectively reduce the overfitting of sample weights and improve the detection rate.

Keywords: Face Detection; Weighting Parameters; AdaBoost Algorithm; Detection Rate

1 Introduction

As an important step of the face recognition, the detection rate of face detection directly affects the results of face recognition. Therefore, face detection gets more and more attention as an independent discipline. Face detection is usually used to detect faces in an input image. If there are faces, fix the position, posture and size of all the faces. Face detection is applied to many aspects broadly, such as authentication, digital video processing and pattern recognition [1].

The methods of face detection can be classified into two categories: the face detection method based on knowledge and the face detection method based on statistical learning. The AdaBoost algorithm proposed by Viola in 2001 is based on Haar-like features [2]. The high detection rate and the rapid detection speed lead the AdaBoost algorithm to apply in wide fields. As an improved Algorithm, the AdaBoost algorithm can adjust the weights of weak classifiers. This algorithm eliminates some unnecessary data, and concentrates on the key data [3].

In the traditional AdaBoost algorithm, training a large number of weak classifiers by changing the sample weights. This method improves the detection rate of face detection. However, when the noise samples and other hard samples are contained by the training samples, these samples are continually classified falsely, so the AdaBoost algorithm will give more and more attention...
to these samples, and the weights of these samples will be very large finally. It leads to the
distortion of sample weights distribution. And the traditional AdaBoost algorithm uses Haar-like
features or LBP features to train the weak classifiers. The number of Haar-like features in a given
image is very large, so training the weak classifiers by Haar-like features takes a long time [4].
And extracting LBP features takes a short time, but it is sensitive to noise [5]. Therefore, these
shortcomings will cause low detection rate for the input image.

In order to solve these problems, this paper proposes the new weighting parameters with chang-
ing weight updating rule and weights normalization rule. The improved weight updating rule is
affected by the detection rate. The improved weights normalization rule is that the face samples
and the non-face samples are respectively normalized. This method can effectively avoid the
overfitting of noise sample weights.

2 The Traditional Adaboost Algorithm

AdaBoost algorithm is an iterative algorithm, its basic idea is to extract the Haar-like features of
the training samples and train the different weak classifiers. Then choose some weak classifiers
and gathering these weak classifiers to form a strong classifier [6]. The traditional AdaBoost
algorithm as follows:

(1) Weights normalization of the training samples. In the traditional AdaBoost algorithm, Nor-
malizing weights after the weights of $m$ face samples and $n$ non-face samples being initialized.
The function of weights normalization is: $\omega_{t,j} = \frac{\omega_{t,j}}{\sum_{j=1}^{m+n} \omega_{t,j}}$. In this function, the weights nor-
malization factor is the sum of all the weights of samples. However, in the training process,
the number of non-face samples is larger than the number of face samples, so the sum of
weights of the non-face samples will be greater than the sum of weights of the face samples
after weights normalization, which will lead to the distortion of weight distribution.

(2) Train the weak classifiers. Extract features for the training samples, and train weak classifier
for each feature: $h_j(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_j \\ 0, & \text{otherwise} \end{cases}$. $h_j(x)$ is the output value of sample $x$ by weak
classifier, $\theta_j$ is the threshold, $p_j$ is the threshold bias, the value only can be $\pm 1$, $f_j(x)$ is the
eigenvalue of sample. The detection rate of weak classifiers trained by Haar-like features is
slightly greater than 50%.

(3) Choose the best weak classifier. After training the weak classifiers, calculate the objective
function $\varepsilon_j = \sum_{i=1}^{m+n} \omega_{t,j} |h_j(x_i) - y_i|$. Select the minimum one of the objective function, and
the weak classifier $h_t$ of the minimum objective function is the best weak classifier.

(4) Update the weights: $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$, when $x_i$ is classified correctly, $e_i = 0$; when $x_i$ is
classified falsely, $e_i = 1$. And $\beta_t = \frac{1}{1-\varepsilon_t}$.

(5) Compose strong classifier. For each training round, make the best weighted sum of weak
classifiers to get strong classifier. Get a strong classifier by the weighted sum of all best weak
classifiers.
The traditional AdaBoost algorithm is easy to shift the focus to the hard samples. Thus, the hard samples have larger weights, which will cause severe distortion of the weight distribution [7]. In order to avoid the overfitting of hard samples, we can change the weight updating rule and weights normalization rule. Then integrate MB-LBP features and Haar-like features into this improved AdaBoost algorithm. The experiment result shows that these methods can reduce the training time and the false detection rate and improve the performance of the face detection system.

### 3.1 Improved weights normalization rule

In the process of AdaBoost algorithm, if the number of face samples is same to the number of non-face samples, the sum of the non-face sample weights may be greater than the sum of the face sample weights after several iterations, which make the weight distribution distort. To prevent weight distribution distorting, we need to give more attention to face samples. This paper proposes an improved weights normalization rule. The face sample weights and non-face sample weights are respectively normalized to ensure more attention on face samples. The improved weight normalization function is as follow:

\[
\omega_{t,j} = \begin{cases} 
\frac{\omega_{t,j} \times m}{\sum_{j=1}^{\omega_{t,j}}}, & y_i = 1 \\
\frac{\omega_{t,j} \times n}{\sum_{j=1}^{\omega_{t,j}}}, & y_i = 0 
\end{cases} 
\]

Where \(m\) is the number of face samples, \(n\) is the number of non-face samples. The number of training samples is: \(q = m + n\). Sample weights are initialized: \(\omega_{1,j} = \frac{1}{m+n} = \frac{1}{q}\).

### 3.2 Improved weight updating rule

In the traditional AdaBoost algorithm, the weight updating rule is:

\[
\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-\epsilon_i}.
\]

The new weight updating rule will make the sample weights classified correctly decrease and make the sample weights classified falsely decrease increase. After several iterations, the sample weights classified falsely continually increase, which will lead directly the false detection rate \(\varepsilon_j\) of weak classifiers generated each round more larger. Therefore, this paper introduces the detection rate \(\gamma\), making the weight updating rule affected by the false detection rate \(\varepsilon_j\) and the detection rate \(\gamma\) concurrently. It can effectively inhibit the overfitting of hard sample weights. The improved weight updating function is as follow:

\[
\omega_{t+1,i} = \omega_{t,i} \times (\beta_t^{1-\epsilon_i})^{(\gamma K)}
\]

Where \(\gamma\) is the detection rate, \(K\) is the regulation parameter, \(\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}\). The new weight updating rule is affected by the detection rate \(\gamma\). The regulation parameter \(K\) can adjust the level of new weight updating rule affected by \(\gamma\). Because of the detection rate \(\gamma \leq 1, \gamma \downarrow \Rightarrow \gamma K \downarrow \Rightarrow \omega_{t,i} \times (\beta_t^{1-\epsilon_i})^{(\gamma K)} \uparrow\). So when the detection rate \(\gamma\) reduces, its updated weight \(\omega_{t,i} \times (\beta_t^{1-\epsilon_i})^{(\gamma K)}\) increases.
And because of \( 0 < \gamma < 1 \), \( 0 < \beta_t < 1 \), it will make the expansion of the sample weights classified falsely decrease, and avoid effectively the overfitting of hard sample weights.

### 3.3 Haar-like features and MB-LBP features fusion

The traditional AdaBoost algorithm is based on Haar-like features. However, because of the large number of Haar-like features and the training time-consuming, this paper integrates MB-LBP features and Haar-like features into the AdaBoost algorithm. The specific linear integration process as follow:

From the weak classifier \( h_j(x) \), which is trained \( T \) times by Haar-like features, determining \( T \) best weak classifiers \( h_1, h_2 \ldots h_T \) after comparing, and the training time of the best weak classifier \( h_T \) is \( T_{ht} \). From the weak classifier \( i_j(x) \), which is trained \( T \) times by MB-LBP features, determining \( T \) best weak classifiers \( i_1, i_2 \ldots i_T \) after comparing, and the training time of the best weak classifier \( i_T \) is \( T_{it} \). The best weak classifiers, which are trained by Haar-like features and MB-LBP features, will be further integration. Given \( h_1, h_2 \ldots h_T \) weight of \( a \), and given \( i_1, i_2 \ldots i_T \) weight of \( b \), \( a + b = 1 \), the best weak classifiers by the two features calculate the objective function (3) and (4), to get the two minimum objective functions, and the output value of \( a \) and \( b \). When the objective function is minimum, the value of \( a \) and \( b \) is weight of the best weak classifiers by Haar-like features and MB-LBP features.

\[
\varepsilon_1 = \sum_{t=1}^{T} a \cdot |h_t - y_t| + \sum_{t=1}^{T} b \cdot |i_t - y_t|
\]

\[
\varepsilon_2 = \sum_{t=1}^{T} a \cdot T_{ht} + \sum_{t=1}^{T} b \cdot T_{it}
\]

### 4 Steps of Improved Adaboost Algorithm

1. Given the training samples \((x_1, y_1), (x_2, y_2) \ldots (x_q, y_q)\), where \( y_i = 0, 1 \) is for the non-face sample and face sample respectively. There are \( m \) face samples and \( n \) non-face samples in the training samples. The number of training samples \( q = m + n \).

2. Initialize sample weights: \( \omega_{1,j} = \frac{1}{m+n} = \frac{1}{q} \).

3. For \( t = 1, 2 \ldots T \):

   a. Normalize the weights: \( \omega_{t,j} = \begin{cases} \frac{\omega_{t-1,j}}{\sum_{j=1}^{m+n} \omega_{t-1,j}} \times \frac{m}{q}, y_i = 1 \\ \frac{\omega_{t-1,j}}{\sum_{j=1}^{m+n} \omega_{t-1,j}} \times \frac{n}{q}, y_i = 0 \end{cases} \)

   b. For each Haar-like feature \( j \), training a weak classifier: \( h_j(x) = \begin{cases} 1, p_jf_j(x) < p_j\theta_j \\ 0, otherwise \end{cases} \)

   is the output value of sample \( x \) by weak classifier, \( \theta_j \) is the threshold, \( p_j \) is the threshold bias, the value only can be \( \pm 1 \), \( f_j(x) \) is the eigenvalue of sample.
(4) Repeat Steps (1), (2), (3). Train the weak classifiers by MB-LBP features and choose the best weak classifier $h_t$.

(5) Integrate $h_t$ and $i_t$, make objective function $\varepsilon_1 = \sum_{t=1}^{T} a \cdot |h_t - y_t| + \sum_{t=1}^{T} b \cdot |i_t - y_t|$ and $\varepsilon_2 = \sum_{t=1}^{T} a \cdot T_{ht} + \sum_{t=1}^{T} b \cdot T_{it}$ minimum.

(6) Output $a$, $b$.

(7) The final strong classifier is:

$$H(x) = \begin{cases} 
1, & \sum_{t=1}^{T} [\alpha_t h_t(x) \cdot a + \alpha_t i_t(x) \cdot b] \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\
0, & \text{otherwise} 
\end{cases}$$

Where $\alpha_t = \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$.

5 Experiment Result

The training samples are chosen from MIT face database, we select 2706 face samples and 4381 non-face samples to extract features. Each sample can be extracted be extracted 400 MB-LBP features or 78,460 Haar-like features. After extraction, use the improved Adaboost algorithm to detect human faces. The testing samples are selected from MIT+CMU face database and the Internet, which contains 1000 face samples, 500 non-face samples and 400 hard samples obtained from the Internet. Then use skin color segmentation to pre-process the given image. The operating environment is AMD3.0GHZ PC, Windows 7 operating system. The experiment is programmed on Matlab7.0.

The experiment is divided into two parts. Firstly, we get the training time and detection rate of the Adaboost algorithm based on Haar-like features, the Adaboost algorithm based on MB-LBP features and the improved Adaboost algorithm in this paper, comparing the training time and detection rate of three methods, where the regulation parameter $K = 0.5$. Then we compare the detection rate and the false detection rate of the testing samples when the value of regulation parameter $K$ is uncertain.

The improved Adaboost algorithm is to select 1353 face samples and 2190 non-face samples from the training samples to extract Haar-like features, and to select the others of the training samples to extract MB-LBP features. These features are used to train the weak classifiers by the improved Adaboost algorithm. Comparing the training time of the Adaboost algorithm based on Haar-like features and the improved Adaboost algorithm., the training time compared is shown in Fig. 1. In Fig. 1, the training time of the improved Adaboost algorithm is obviously less than the traditional Adaboost algorithm.

The detection rate of the Adaboost algorithm based on Haar-like features, the Adaboost algorithm based on MB-LBP features and the improved Adaboost algorithm is shown in Fig. 2. In the
improved Adaboost algorithm, when the number of best weak classifiers is small, the detection rate is small too. With the increase of the number of the best weak classifiers, the detection rate is gradually increased and eventually exceeds the traditional Adaboost algorithm.

In this paper, we can use the improved Adaboost algorithm to train 110 best weak classifiers, these weak classifiers are applied to compose the strong classifier, and the strong classifier detects the testing samples. When the value of the regulation parameter $K$ is different, the detection rate and the false detection rate are also different. The value of the regulation parameter $K$ respectively is 0.1, 0.3, 0.5 and 0.7. The detection rate is shown in Fig. 3. When the number of best weak classifiers is small, the detection rate is also small. With the increase of the number of
of the best weak classifiers, the detection rate is gradually increased. Eventually, when $K = 0.1$, the detection rate is little lower. When the value of $K$ is 0.3, 0.5 or 0.7, the detection rate is basically the same.

When the value of $K$ is 0.1, 0.3 and 0.5, the false detection rate is shown in Fig. 4. While the number of the best weak classifiers is small, the false detection rate is large. With the increase of the number of the best weak classifiers, the false detection rate gradually decreased. The false
detection rate of $K = 0.1$ is significantly greater than the false detection rate of 0.3 and 0.5. And the false detection rate ultimately is about 2%.

6 Conclusions

In this paper, the parameters rule is made improvement based on the traditional AdaBoost algorithm. We define the new weights normalization rule and weights updating rule, and use MB-LBP features and Haar-like features fusion instead of Haar-like features, then apply to the AdaBoost algorithm to detect human faces. The experiment result on MIT+CMU test set shows that this method can effectively reduce the false detection rate and improve the detection rate, and prevent the overfitting of sample weights. In recent years, many researchers [8] have made a lot of improvements of the AdaBoost algorithm, the research results showed that this improvement can detect human faces preferably. Referring to this method, how to detect multi-angle faces in a given image is a novel direction in the future research.

References