Application of Ear Detection in Non-frontal Face Recognition

Yingjie Luo¹, Tianfei Shen²

¹Department of Automation, Shanghai University, Shanghai, Province, 200444, China
²Department of Automation, Shanghai University, Shanghai, Province, 200444, China

*Corresponding author’s e-mail: 18717886785@163.com

Abstract. Face gesture recognition is a research hotspot in the field of machine vision and pattern recognition. In this paper, we use the Haar-Like feature-based Gentle Adaboost algorithm to detect the ear of non-frontal face. An ear sample library for ear detection of non-frontal face images was constructed. Through zoom detection window and determining the ROI area, it can successfully detect ear and work better than the basic Adaboost algorithm. The detection rate of the algorithm in the experiment reached 82%. It is suitable for most of the non-frontal faces we photographed. It still has a good detection effect for images with a small amount of hair or eyeglass frame interference. Based on the contour information of the ear, an approximate method for determining the lower end of the ear based on edge detection and a head and neck boundary determination algorithm based on the lower end of the human ear are proposed, which is more complete than the previous algorithm based on the determination of the concave points on both sides of the neck. The face area is segmented to make the face contour finally applied on the fitted ellipse more accurate.

1. Introduction
Face gesture recognition is a research hotspot in the field of machine vision and pattern recognition, which has attracted the attention of a large number of researchers. However, a lot of researches on face gesture recognition are mainly directed at positive faces, and there are few studies on non-frontal face gesture recognition. In fact, non-positive facial gesture recognition has great practical significance in many life scenes. Through the analysis of the five kinds of gesture images of forward, left, right, up and down, it is found that some geometric features of the human ear can play a certain auxiliary role in the determination of the non-positive facial momentum. The application of ear geometry information and position information can improve the performance of the recognition system. Therefore, this article focuses on the detection of human ears for non-frontal faces.

At present, there are many research methods at home and abroad for the ear detection. In general, ear detection methods mainly include methods based on skin color models and basic points of structural features and methods based on machine learning algorithms. In 2002, Victor B. applied principal component analysis (PCA) technology to the ear detection[1], which achieved a good result. Scale-invariant feature transform (SIFT) technology was first applied in this field[2]. In 2006, Hui Zhao proposed a real-time ear tracking method based on skin color and contour information[3]. In 2007, Wei Zhang proposed a fast ear detection algorithm based on Adaboost algorithm[4]. In 2014, based on Non-negative sparse representation, Baoqing Zhang proposed a new algorithm of blocked ear recognition[5]. In 2017, Long Chen proposed a single-sample human ear recognition method in uncontrolled scenes[6].

Since the images collected in this paper are non-frontal, the ear information is relatively incomplete, and it is more suitable for the ear detection method based on the machine learning
algorithm. Therefore, through zoom detection window and determining the ROI area, an Adaboost-based machine learning algorithm is used to detect ears in this paper. Based on the contour information of the ear, an approximate method for determining the lower end of the ear based on edge detection and a head and neck boundary determination algorithm based on the lower end of the human ear are proposed.

2. Gentle adaboost algorithm based on Haar-Like features

Proposed by Yoav Freund and Robert Schapire\cite{7}, the Adaboost algorithm combines multiple weak classifiers into strong classifiers to achieve classification purposes, which is an adaptive boosting algorithm. It has a wide range of applications, and there are many different evolutionary algorithms for different fields. Discrete Adaboost, Real Adaboost, Gentle Adaboost algorithm and so on are Commonly used in face detection systems. The experiment shows that Gentle Adaboost performs better in face detection\cite{8}. Gentle Adaboost was first proposed by Friedman\cite{9} and evolved on the basis of Discrete Adaboost and Real Adaboost. The weight update during the training process of this algorithm is smoother than other Adaboost algorithms, which can avoid the occurrence of overfitting. Therefore, the Gentle Adaboost algorithm is used to detect ears of non-positive faces in this paper.

2.1. Haar-Like feature

The Haar-Like feature was proposed by Papageorgiou C.\cite{10}. Using diagonal features, Rainer Lienhart and Jochen Mayd extended the Haar-Like feature library\cite{11}. Because it can describe the grayscale change of the image, it is very effective in scenes with rich light and dark information. Because of the obvious grayscale change characteristic of the ear, it is very suitable for feature collection using several templates in the Haar-Like feature (horizontal, vertical, diagonal, and circle center), which is shown in figure 1.

![Figure 1. Haar-Like feature](image)

2.2. Integral graph algorithm of Haar-Like feature

According to the calculating formula of feature number\cite{12} and the Haar-Like feature concept, a 24x24 image can produce more than 160,000 feature values. Selecting the image area feature and recalculating them each time, it spends a lot of time on calculation. Therefore, the integral graph algorithm is used in this paper, which is proposed by Rainer Lienhart\cite{12}.

The integral graph refers to the grayscale integral value of a certain point in the image, which is equal to the sum of all grayscale values in the area from the image start point to the end point. According to the Haar-Like characteristic attribute, the calculation of integral graph is divided into two cases: the inclination angle is 0\degree and the inclination angle is 45\degree\cite{13}. With this method, only a small amount of calculation is needed to obtain the "integral graph", which can calculate rectangular feature values with different scales in a short time, thus greatly improving the calculation speed.

2.3. Composition of multi-layer classifier

2.3.1. The weak classifier
The basic Haar-Like features are used to build a weak classifier. The formula is shown as following.

$$h(x, f, \theta) = \begin{cases} \alpha_1 & \text{if } f(x) < \theta \\ \alpha_2 & \text{others} \end{cases}$$  \hspace{1cm} (2-1)

In this formula, $h$ is the weak classifier and $x$ represents the detection target. $f$ is the Haar-Like feature. $\theta$ means the threshold of the current weak classifier. $f(x)$ is the corresponding feature value of the feature $f$ in the detection target $x$. $\alpha_1$ and $\alpha_2$ are the detection result, that is, the confidence that the current detection area is the human ear. In this paper, +1 is defined as a human ear sample, and -1 is a non-human ear sample. The range of values is $\{\alpha\} \leq 1$. When all samples are correctly classified, $h(x, f, \theta) = +1$. In contrast, all samples are incorrectly divided, $h(x, f, \theta) = -1$.

The obtained feature threshold value is compared with the input image feature value to determine the confidence, which can well avoid the occurrence of overfitting. It is determined by the sample characteristic value size and the mean square error of the sample identification. The weak classifier with the smallest total mean square error is the optimal weak classifier.

2.3.2. The Strong classifier
Because of the poor recognition only by the weak classifier, the weak classifier needs to be combined by the strong classifier to complete the task of classifying and identifying the target.

The formula of the strong classifier is shown as following.

$$H(x, \varphi) = \begin{cases} 1 & \sum_{i=1}^{T} h_i(x) \geq \varphi \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (2-2)

In this formula, $H$ is the strong classifier, $x$ is the detection target, $\varphi$ is the threshold of the strong classifier, and $H(x, \varphi)$ is the detection result of the strong classifier. $h(x)$ is the detection result of the weak classifier and $T$ represents the number of weak classifiers that make up a strong classifier. When $H(x, \varphi) = 1$, the target is detected.

According to the setting minimum positive detection rate and maximum false positive rate, the strong classifier is determined. In Each iteration, a weak classifier is added to the strong classifier. Later the strong classifier threshold is adjusted to make the positive detection rate meet the minimum positive Detection rate. And then the sample weight distribution is adjusted according to the classification result of the weak classifier. So the loop ends until the false positive rate is less than the maximum false positive rate. And the strong classifier is obtained.

$$\text{positive detection rate}(\text{rec})= \frac{\text{Number of positive samples correctly detected}}{\text{Total number of positive samples}}$$

$$\text{false positive rate}(\text{fault})= \frac{\text{Number of negative samples misdetected as positive samples}}{\text{Total number of negative samples}}$$

2.3.3. The cascade classifier
The cascade classifier consists of strong classifiers connected in series. It can be seen that as long as there is a strong classifier that determines that the current detection window is a non-human ear area, the area is directly determined as a non-human ear area. So that the strong classifier in front of the cascade structure can filter out a large number of non-human ear areas. Only the regions detected by all strong classifiers can be regarded as human ear regions. This cascade structure reduces the false positive rate while increasing the detection speed.

In the training of cascade classifier, each iteration adds a strong classifier to the cascade classifier. When the number of cascades meets the set condition, the false positive rate of the cascade classifier is
calculated. So the loop ends until the number of cascades of the classifiers reaches the set maximum value or the false positive rate of the cascade classifier reaches the set requirements.

3. Ear detection

3.1. The ear samples
The non-frontal facial momentum pictures collected in the experimental environment are manually marked and cropped to construct a 200-sample human ear original sample library, which includes each 40 ear images of forward, upward, downward, left, and right. In order to improve the number and quality of samples, the sample data was enhanced by horizontal flip and angular rotation.

The final total number of positive samples is 1080. After processing by the sample gray-scale world balance algorithm and the size normalization adjustment, the sample size is 30x30 pixels, as shown in Figure 2. Considering that the current research is limited to the laboratory environment, the non-human ear environment images under the same environment are used as negative samples. There are a total of 3000 negative samples for learning and training.

3.2. The process of classifier training
we use Opencv and Python for code writing and classifier training. Among 1080 positive samples and 3000 negative samples, each strong classifier iteration selects 800 positive samples and 2000 negative samples for training. We set the maximum false positive rate of each strong classifier to 0.5 and the minimum positive detection rate to 0.995. And the false positive rate of the final cascade classifier should be set to be less than. The maximum number of cascade strong classifiers is 20. This training takes a total of 13.5h, and the generated cascade classifier is composed of 16 strong classifiers.

3.3. The process of ear detection

3.3.1. Detection window zoom
The detection image is fixed and the detection window is sequentially zoomed as a certain ratio. Detection window after each zooming is subjected to sliding detection at a certain step to traverse the detection image. Because the original image size is fixed, the integral graph calculation only needs to be performed once. The original training window size is adapted directly to the scaled detection window size, and the Haar-Like feature value is calculated to complete the detection of the target area. Compared with the method based on the detection image scaling, the detection method requires a small amount of calculation and fast detection speed, and the scaling ratio is set to 1.1 for multi-scale detection.
Through this method, a large number of overlapping candidate areas appear in the target area as shown in Figure 3(a). Therefore, it is also necessary to merge the overlapping candidate regions on the detection results and delete the false detection regions according to the number of overlapping candidate regions. In this paper, the parameters that define each square candidate area are represented by \((x, y, s)\), where \(x\) and \(y\) are the coordinates of the upper left corner of the rectangle, and \(s\) is the side length of the candidate area (a 30x30 detection window is used in training, so the output is a square candidate area). According to the relative distance of the upper left corner coordinate point of the adjacent candidate area and the size of the side length as the judgment condition, the overlap judgment is performed. Assume the parameters of the two candidate areas are \((x_1, y_1, s_1)\) and \((x_2, y_2, s_2)\). If the following three conditions are met, it is judged as an overlapping area, and the area is merged:

1. \[s_1/1.2 \leq s_2 \leq s_1 \times 1.2\];
2. \[x_1 - 0.2x_1 \leq x_2 \leq x_1 + 0.2x_1\];
3. \[y_1 - 0.2y_1 \leq y_2 \leq y_1 + 0.2y_1\].

As shown in Figure 3(a), there are a large number of overlapping regions in the target area. The false detection candidate regions generally only appear in a certain detection window scale, so the false detection candidate regions have only a few or no overlapping regions. According to this feature, more than 5 overlapping candidate regions are set in this paper to be merged and output as the final target region. The detection effect is shown in Figure 3(b).

3.3.2. Determine the ROI area

Because non-skin areas are detected wrongly as shown in Figure 3(a), the skin color model\(^{14}\) is used to locate the image ROI area, which is shown in Figure 4(b). It improves the accuracy of ear detection. Since the skin color model is part of the previous image processing flow, only the corresponding image area needs to be extracted. Determining the ROI area reduces the image area calculated by the Adaboost algorithm, shortens the time required for human ear detection, and reduces the false detection rate.
4. Neck Rejection

In the previous neck rejection algorithm, the determination of the inflection points on both sides of the face contour is not accurate enough. Part of the neck area is misinterpreted as the face area. The ellipse fitted by the face contour sometimes contains the neck area. The area affects the geometric features of the face ellipse, thereby reducing the effectiveness of the face gesture recognition algorithm. Therefore, this paper obtains the chin position by determining the position of the human ear.

The steps of the neck removal algorithm designed in this paper are as follows:

1. Extract the part of the human ear obtained by the human ear detection algorithm, binarize the image, and remove the image interference in non-skin areas, as shown in Figure 5(b);
2. Canny operator edge detection, a relatively complete profile of the ear is obtained, as shown in Figure 5(c);
3. According to the position of the human ear relative to the face, determine whether the left ear or the right ear is detected. If the right ear is detected, we get the momentum to the left. If it is the left ear, intercept the lower left part of the outline of the ear, as shown in Figure 5(d). Select the minimum value of the y-axis, which is the lower end of the human ear;
4. Cut the contour curve of skin color obtained before\cite{14}. According to the empirical knowledge of face structure, the position of the chin is generally less than 1/2 to 1/3 of the contour height, so the left half of the contour curve is intercepted, as shown in Figure 5(e). The maximum value of the y-axis is selected as the other point;
5. Connect the obtained two points, as shown in Figure 5(f);

![Figure 5. Neck Rejection](image-url)
5. Experiment result

5.1. The results of ear detection

The final algorithm input image is converted from 640x480 pixels to 260x260 pixels. The amount of calculation is greatly reduced, and the detection speed is significantly improved. At the same time, the interference of non-skin areas on detection is reduced, with reducing the false detection rate.

![Ear detection results](image)

(a) Downward  (b) Forward  (c) Upward  (d) Heading to the right  (e) Heading to the left

Figure 6. The results of ear detection

50 pictures that did not participate in the training are used to test, including 10 pictures of each of the five kinds of states.

| States          | Forward | Upward | Downward | Left  | Right |
|-----------------|---------|--------|----------|-------|-------|
| Detection rate  | 90%     | 80%    | 80%      | 60%   | 100%  |
| False detection rate | 0%    | 11.1%  | 0%       | 14.3% | 9.1%  |

Table 1 shows that the detection rate of the forward and right images is relatively high at 90%. The detection rate of the left-handed momentum is the lowest, only reaching 60%, because But it doesn’t matter. Only under this kind of condition, the right ear appears. It is easy to distinguish. Regardless of the left state, the detection rate of this algorithm is 87.5%, and the false detection rate is 5.05%. It has wide applicability and can deal with interference in the experimental environment, but the training time is long and the detection effect is greatly affected by the number of samples.

The comparison between the method of this paper and the basic Adaboost algorithm for the detection of 50 experimental samples that did not participate in training is shown in the following table. It can be seen that the false detection rate has been greatly reduced, and the detection cost It is almost 50% less than the basic method.

| Index                        | Basic Adaboost detection algorithm | The new algorithm |
|------------------------------|------------------------------------|-------------------|
| Detection rate               | 80%                                | 82%               |
| False detection rate         | 28.6%                              | 6.8%              |
| Average time                 | 24.16ms                            | 12.56ms           |
5.2. Contrast experiment of ellipse fitting between two head and neck boundaries

In this paper, based on the previous head and neck segmentation algorithm\textsuperscript{[14]}, a head and neck segmentation algorithm based on ear position is proposed. The recognition result is shown in Figure 7. The first picture of each group is the original image, and the second picture is the face ellipse fitting effect of the head-neck boundary segmentation algorithm based on the neck concave points. The three figures are the face ellipse fitting effect of the head-neck boundary segmentation algorithm based on the ear position and the neck concave point.

Comparison of multiple experiments shows that the new algorithm can segment the head and neck region more accurately under various situations than the previous algorithm. It plays an important role in the correct face Ellipse fitting.
6. Conclusion
In this paper, an ear sample library for ear detection of non-frontal face images was constructed. The Haar-Like feature-based Gentle Adaboost algorithm is used to detect the ear of non-frontal face. Through zoom detection window and determining the ROI area, it can successfully detect ear and work better than the basic Adaboost algorithm. It can accurately detect the position of the left and right ears. The detection rate of the algorithm in the experiment reached 82%. It is suitable for most of the non-frontal faces we photographed. It still has a good detection effect for images with a small amount of hair or eyeglass frame interference.

Based on the contour information of the ear, an approximate method for determining the lower end of the ear based on edge detection and a head and neck boundary determination algorithm based on the lower end of the human ear are proposed, which is more complete than the previous algorithm based on the determination of the concave points on both sides of the neck. The face area is segmented to make the face contour finally applied on the fitted ellipse more accurate.

In this paper, the experimental samples and the ear sample library are manually collected and manually marked. The laboratory samples are seriously insufficient due to various restrictions, and the quality of the collected samples is also uneven. The number and quality of the samples seriously affect the training quality of the human ear detection classifier and the final momentum recognition classifier, which is one of the reasons why the detection rate of the detection algorithm used in this article can only reach 82%. How to improve the number of samples and the quality of samples is the key to further research.

Acknowledgments
Authors wishing to acknowledge assistance or encouragement from colleagues, special work by technical staff or financial support from organizations should do so in an unnumbered Acknowledgments section immediately following the last numbered section of the paper.

References
[1] Barnabas V., Kevin B., Sudeep S. (2002) An evaluation of face and ear biometrics[C], Pattern Recognition, 16th International Conference on IEEE, vol. 1: 429-432.
[2] David J. Hurley, Mark S. Nixon, John N. Carter (2005) Ear biometrics by force field convergence[C], International Conference on Audio-and Video-Based Biometric Person Authentication. Springer, Berlin, Heidelberg, 386-394.
[3] Zhao W., Mu Z.H., Liu K. (2006) Real-time human ear tracking based on skin color and contour information[J], Journal of Image and Graphics, 11(7): 949-935.
[4] Zhang W., Mu Z.H., Yuan L. (2007) Human Ear Detection and Tracking Based on Improved AdaBoost Algorithm[J], Journal of Image and Graphics, 12(2): 222-227.
[5] Zhang B.Q., Mu Z.H., Zeng H. (2014) Occlusion Human Ear Recognition Based on Non-negative Sparse Representation[J], Journal of Computer-Aided Design and Graphics, 26(08): 1339-1345+1353.
[6] Chen L. (2017) Research on Single Sample Human Ear Recognition in Uncontrolled Scenes [D], University of Science and Technology Beijing.
[7] Freund Y., Schapire R. E. (1995) A decision-theoretic generalization of online application to boosting[J], Journal of Computer and System Science.
[8] Lienhart R., Maydt J. (2003) Empirical analysis of detection cascades of boosted classifiers for rapid object detection[J], Pattern Recognition, 297-304.
[9] Friedman J., Hastie T., Tibshirani R. (2000) Additive logistic regression: a statistical view of boosting[J], Annals of Statistics, 28(2): 337-407.
[10] Papageorgiou C. P., Oren M., Poggio T. (2002) A General Framework for Object Detection [C], International Conference on Computer Vision. IEEE.
[11] Lienhart R., Maydt J. (2002) An Extended Set of Haar-like Features for Rapid Object Detection[C], Image Processing. 2002 International Conference on IEEE.
[12] Viola P., Jones M. (2001) Rapid Object Detection using a Boosted Cascade of Simple Features[C], IEEE Computer Society.

[13] Cao E.K. (2003) Research on Face Detection Algorithm Based on Gentle Adaboost [D], Xi’an: Master Degree Thesis of Xidian University.

[14] Wang H.N. (2018) Research on Non-Front Face Contour Extraction Method under Complex Environmental Conditions [D], Shanghai: Master Degree Thesis of Shanghai University.