Efficient Face/Pose Detection Based on Machine Learning

Shwu-Huey Yen, Chin-Wei Tsai, Tai-Kuang Li
Department of Computer Science and Information Engineering
Tamkang University, Taipei, Taiwan, R.O.C
shyen@cs.tku.edu.tw, 693191123@s93.tku.edu.tw, 695410141@s95.tku.edu.tw

ABSTRACT

Machine learning is a state-of-the-art scheme in solving many kinds of complicated problems. This paper utilizes two types of machine learning algorithms to detect skin and face/pose respectively. Initially a hierarchical neural network is applied for skin detection. Begin with a neural network to overcome the diversity of light and follow by a second neural network to get over colors near the skin color. After the skin area is detected, an AdaBoost learning algorithm is implemented for face/pose detection. Haar-like features [11][12] are utilized as features of modified Adaboost to determine whether there is a left, frontal, right, or non-face in a 20 x 20 sliding window. Experimental results show that the proposed method achieves a good performance in skin color detection, capacity of coping with the problems of scaling, rotation and multiple faces, as well as a good detection rate.

1: INTRODUCTION

Face detection is a must preprocess to many applications, such as the surveillance system, nursing system, driver statement analysis, etc.. Because of the variation of light, the difference of races, genders, ages, and unconstraint of the background, face detection is still a challenging problem with a long history [1]-[17]. There are a lot of methods for face detection, and some of common approaches are briefly introduced below.

- Feature-based: determine a face according to features like shape[9], edge[10], etc..
- Knowledge-based: utilize the face's features like eyes, nose, or mouth to find a face.[13]
- Template-based: use pre-defined face templates like eyes, mouth, or ellipse template to measure the similarity.
- Machine learning-based: machines learn the rules from provided training samples and classify the test sample accordingly [15][16][17]. Especially Viola and Jones [11][12] proposed a rapid object detection system which utilized Adaboost to train an efficient classifier for face detection.
- Skin color extraction: locate skin area first and determine if there is any faces. In this way, the system can be more efficient and accurate since it does not need to search the whole image.

In the proposed system, skin areas are detected first by neural network and then face/pose detection is implemented by Adaboost algorithm in the detected skin areas. In Section 2, we will review related work on skin color extraction as well as Adaboost algorithm. Section 3 describes the frame work of the proposed system. The experimental results and discussion are given in Section 4. Finally conclusions and future work are given in Section 5.

2: SKIN COLOR DETECTION AND ADABOOST

Choose an appropriate color space is the first step for correctly extracting skin areas [1]. In here, three common color models RGB, YCbCr, and HSV are discussed. Adaboost learning algorithm is also introduced.

2.1: COLOR SPACE FOR SKIN COLOR

RGB is composed of Red, Green and Blue three color components which is widely used in digital media. The advantage of RGB color space is calculation efficient, but it is sensitive to light and causes false detection. There are quite a few research try to locate skin area by defining thresholds on RGB models. For example, [2]-[4] found skin pixels cluster in a small region in RGB color space. We list one of these rules [3] in Eq. (1) for skin pixels detection as a comparison later to our work in Section 4.

Contrast to RGB, YCbCr color space separates luminance Y from color and converts blue and red into CbCr values. According to the strength of Y($Y>128$ or $Y \leqq 128$), [6] gives different skin color thresholds respectively as in Eq. (2). HSV is also a color space that separates luminance from color information. One of these rules [18] for skin pixels detection is given in Eq. (3) as a comparison later to our work.

$$\left.\begin{array}{l} R > 95,\ G > 40,\ B > 20, \\ \mathrm{Max}(RGB) - \min(RGB) > 15 \\ \mathrm{abs}(R-G) > 15 \\ R > G,\quad R > B \end{array}\right\} \qquad (1)$$

$$
\left.\begin{array}{l}
if\,(Y > 128) \\
\theta_1 = -2 + (256 - Y)/16, \\
\theta_2 = 20 - (256 - y)/16, \\
\theta_3 = 6, \theta_4 = 8 \\
elseif\ \ (Y <= 128\ ) \\
\theta_1 = 6, \theta_2 = 12\ , \\
\theta_3 = 2 + Y / 32\ , \\
\theta_4 = -16 + Y / 16 \\
C_r \geq -2(C_b + 24),\ C_r \geq -2(C_b + 17), \\
C_r \geq -4(C_b + 32),\ C_r \leq (220 - C_b)/6, \\
C_r \geq 2.5(C_r + \theta_1),\ C_r \leq 4(\theta_2 - C_b)/3, \\
C_r \geq 0.5(\theta_4 - C_b),\ C_r \geq \theta_3
\end{array}\right\} \quad (2)
$$

$$0.23 \leqq S \leqq 0.68, \quad \text{and} \quad 0 \leqq H \leqq 50 \quad (3)$$

2.2: ADABOOST ALGORITHM

A brief introduction of Adaboost algorithm [11][12] is given in the Fig. 1. When the strong classifier H composed of T features (weak classifiers) has been trained by the provided training samples, it will evaluate H(x) for every candidate x. If H(x) = 1 then x is classified as a positive and a negative otherwise. The confidence value for candidate x, $C_H(x)$, is defined as in Eq. (4) which is the value evaluated in H(x) indicating how similar the features in H with those in x.

The confidence value $\quad C_H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \qquad (4)$

The proposed system is not only able to detect whether there is a face, but also determine its pose (a left, frontal, or right face). Adaboost algorithm is modified to implement 3- dimension vector y_i such that (1, 0, 0) (or (0, 1, 0), (0, 0, 0)) for x_i to be a left face (or frontal-face, non-face), etc.. The details of training each weak classifier h_j for feature j is explained in the following.

Assuming there are n features (weak classifiers), a look up tables with n bins will be built such that each feature bin has 3 classifiers, $h_j(v_1, v_2, v_3)$, $v_1, v_2, v_3 \in \{1, 0\}$ representing left & non-left, frontal & non-frontal and right & non-right face, j = 0, …, n-1. To train a classifier, for example, the frontal face classifier for feature j, i.e., to determine the threshold of v_2, we evaluate $f_j(x)$ for every training sample x where f_j is the function for feature j. Instead of having only one hard threshold as usually do in training weak classifiers of Adaboost, we adopt the method in [19] to have a better judgment. First, values of $f_j(x)$ for all x are normalized and evenly divided into k intervals. For those values $f_j(x)$ fall in the interval t, t = 0, 1,…, k-1, find the average of $f_j(x)$ from positive samples (i.e., x is a frontal face) and negative samples (i.e., x is not a frontal face) respectively. Next, take the midpoint of the two average values as the threshold for the interval t of the frontal face classifier of feature j. By this way, a look up table can be set up, i.e. $h_j(v_1, v_2, v_3)$ is determined for every feature j.

To determine whether a sample x is a frontal face according to the feature j, referring to the threshold of the interval containing $f_j(x)$ for frontal face classifier on the bin h_j of the look up table, an output of v_2 in h_j will be 1 if $f_j(x)$ is belonging to the same half with positive training samples and 0 otherwise. Similarly for determining whether a sample x is a left or right face according to the feature j. Although the size of the look up table is large, n x 3 x k, but it is set up during the training stage, when determining the status of a test sample it only needs to refer to the table which is very time efficient.

- Given n training images(x_i, y_i) where i = 1, …, n, and $y_i \in \{0,1\}$ for negative and positive training samples respectively.
- Initialize the weights $\omega_{1,i} = \frac{1}{2m}$ or $\frac{1}{2l}$ for y_i = 0,1 respectively, where m and l are the number of negative and positive training samples.
- For t = 1…T:
    - For each feature j, a weak classifier h_j is trained using $\omega_{t,i}$. Next, calculate the error: $$\varepsilon_j = \sum_i \omega_{t,i} \left| h_j(x_i) - y_i \right|.$$
    - let $h_k(\cdot)$ to $h_t(\cdot)$ if for $\forall j \neq k,\ \varepsilon_k < \varepsilon_j$, i.e. to choose the weak classifier $h_t(\cdot)$ with the lowest error). Let $\varepsilon_t = \varepsilon_k$.
    - Update $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$ and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ where e_i = 0, 1 for training sample x_i being correctly or incorrectly classified by $h_t(\cdot)$.
    - Normalize ω_{t+1} so that it is a weight distribution.
- The final strong classifier is: $$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$ where $\alpha_t = \log \frac{1}{\beta_t}$.

Fig. 1. The AdaBoost Algorithm

3: THE PROPOSED SYSTEM

The proposed system utilizes two types of machine learning algorithms to detect skin and face with pose respectively. Initially a hierarchical neural network is applied for skin detection. Begin with a neural network to overcome the diversity of light and follow by a second neural network to make a distinct for colors near the skin color. After skin areas are detected, some morphological operation and simple connected component analysis are applied to eliminate possible noises. Finally, every connected component of skin area will be fed into the trained Adaboost algorithm for face & pose detection.

3.1: THE DETECTION OF SKIN AREAS

In the propose system, only the detected skin area will be further processed for face detection. Due to hard thresholds of skin color as in Eq. (1), (2), (3), some skin color pixels may be sacrificed and this causes difficulties in later steps. A hierarchical neural network is thus designed to achieve the best of both tasks, i.e., preserve the skin area and eliminate non-skin pixels.

The influence of light variation on colors is one of the main reasons that makes skin color detection a challenging task. To overcome this problem, the first neural network is trained separately according to the strength of luminance Y ($Y>128$ or $Y \leqq 128$). Due to the nature of connectedness of skin pixels, the neural network takes cross shape features on YCbCr color space as shown in Fig. 2. For any pixel, together with 8 other pixels as indicated (2 on its top, bottom, left, and right), each with Y, Cb, Cr 3 values, 27 values are the input for the neural network.

		YCbCr		
		YCbCr		
YCbCr	YCbCr	YCbCr	YCbCr	YCbCr
		YCbCr		
		YCbCr		

Fig.2. The cross feature taken for first neural network

Observing the candidate skin areas output from the first neural network, although all skin pixels are preserved, there are a few similar color non-skin pixels are kept as well. Therefore, the output of the first neural network will be processed again in the second neural network. The goal of the second neural network is to eliminate those non-skin pixels but have color similar to skin color. It takes features R, G, B values on RGB color space as input.

3.2: THE DETECTION OF FACES AND POSES

After skin areas being located, morphological opening and closing are applied for eliminating noises. Also a skin area will be discarded if the proportion of width and height of the connected component of a skin area is larger than 4 or smaller than 1/4, or any of height or width is less than 20 pixels.

The training samples for AdaBoost are 20 x 20 images of left, frontal, right, and non-face taken from websites and the CVL face database [20]. These training images are manicured to cover only facial features as much as possible. The method of bootstrap on training samples is applied to promote the performance of the AdaBoost. Testing images are downloaded from websites as seen in Fig. 4, 5, 6 in Section 4. The Haar-like features, as shown in Fig. 3, and the variances of first three Haar-like features are adopted for features in Adaboost.

To determine whether a skin area containing any faces, a sliding window of 20 x 20 is applied on every connected component of skin area, from left to right and top to bottom. If the center portion (10 x 10) of a sliding window contains less than 85 skin pixels, then this sliding window will be skipped and go to next window. These sliding windows are fed into the AdaBoost algorithm one by one for determination. To detect all sizes of faces, the process is repeated by a scale of 0.8 on the image until its height or width is less than 40. Since every skin area will be examined repeatedly on different scales, it is very possible that a face is detected more than once. The confidence value in Eq. (4) will be used as the criterion. When there are overlapping windows with positive response of the same type (left, frontal, or right face), reserve the window with largest confidence value and eliminate the rest.

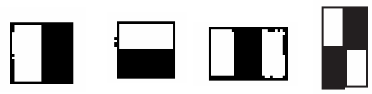


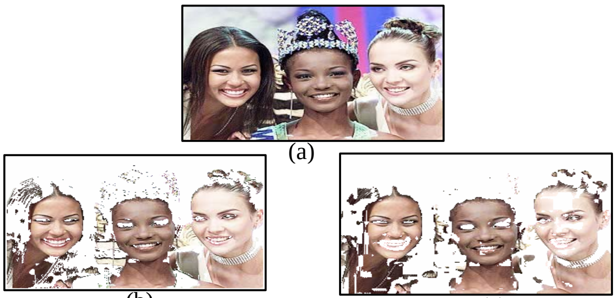
Fig. 3. The Haar-like features

4: THE EXPERIMENTAL RESULTS

Some experimental results are shown and discussed here. For skin detection, as shown in Fig. 4, the original image (a) is affected by green color on the lower portion and the light on the face is uneven too. Our method Fig. 4 (e), has the best result among all. As notice, the method of YCbCr [6], Eq. (2), also performs well compared with Fig. 4 (b), (c). In fact, [6] in general shows satisfying results and it is referenced by other research frequently when skin color detection problem is discussed. Thus, in Fig. 5 & 6, only [6] will be compared with our result. In Fig. 5, testing on different races, observing (b) & (c), both methods can extract most of skin areas. Our method preserves skin areas more, for example, the forehead of the lady on the right, with the price that some non-skin pixels are kept as well, as the left shoulder of the lady on the right. Same consequence is derived on Fig. 6. Due to the similarity to skin color of colors on background and the lady's hair, our method, Fig. 6(b), preserves not only correct skin area but hair and background too. As the [6]'s method, Fig. 6(c), wrongly identifies blond hair as skin too but it successfully eliminates the background with the price that it also eliminates facial skin area. As a preprocess of face detection, Fig. 6(c) has no face area kept at all which consequently results in no face detected. Therefore, our method is more suitable for later face detection.

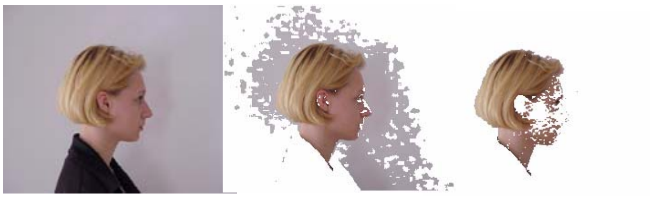
(a) (b) (c) (d) (e)

Fig.4. Results of skin detection with (a) the original image, and by methods of (b) HSV [18], (c) RGB [3], (d) YCbCr [6], (e) ours.



(a) (b) (c)

Fig.5. Results of skin detection on different races with (a) the original image, and by methods of (b) ours, (c) YCbCr [6].



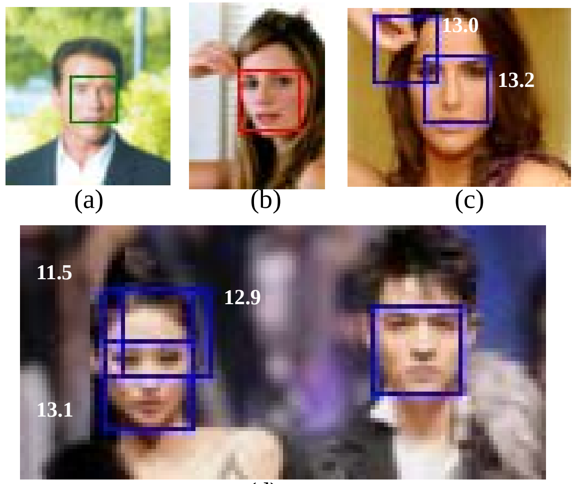
(a) (b) (c)

Fig.6. Results of skin detection with (a) the original image, and by methods of (b) ours, (c)YCbCr [6].

As face/pose detection, Fig. 7 shows some of our experimental results. The red, blue, green boxes are for detected left, frontal, right faces respectively. These images, except (c), are natural images with all kinds of background setting. Our method in general shows satisfying results. In (c) and (d), there are multiple boxes with confidence values indicated, the one with the largest value will be the representative box which also is the correct face area.

5: CONCLUSION

In this paper, we use a hierarchical neural network for skin detection. Begin with a neural network to overcome the diversity of light and follow by a second neural network to get over colors near the skin color. After the skin area is detected, an AdaBoost learning algorithm is implemented for face/pose detection. Experimental results show that the proposed method achieves a good performance in skin color detection and face/pose detection, capacity of coping with the problems of scaling, rotation and multiple faces.



(a) (b) (c)

(d)

Fig.7. Results of face detection where red, blue, green boxes are for left, frontal, right faces respectively.

The difficulty of skin detection and face/pose detection lies on unconstrained background and diversity of the target. By machine learning to find subtle distinctions among positive and negative samples is a promising resort and the success of machine learning largely depends on training samples. Thus how to choose enough and good training samples is an interesting problem. In the future, we will focus on finding better training samples and possibility of integrating the system with other learning methods, such as SVM and PCA.

REFERENCES

[1] Son Lam Phung, Abdesselam Bouzerdoum, Douglas Chai,"Skin Segmentation Using Color Pixel Classification: Analysis and Comparison," IEEE Trans.on pattern analysis and machine intelligence, Vol. 27, No. 1, January 2005.

[2] J. Yang, A. Waibel, "Tracking human faces in real time," CMU-CS-95-210, 1995.

[3] Franc Solina, Peter Peer, Borut Batagelj, Samo Juvan, Jure Kova c, "Color-based face detection in the 15 seconds of fame art installation," Proceedings of Mirage 2003, INRIA Rocquencourt, France, March, 2003.

[4] Kah Phooi seng, Andy suwandy, L. M ang, "Improved automatic face detection technique in color images," IEEE, 2004.

[5] Yanjiang Wang, Baozong Yuan, "A novel approach for human face detection from color images under complex background," Pattern Recognition 34 (2001) pp. 1983 – 1992.

[6] Linhui Jia and L. Kitchen, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis," IEEE Transactions on Image Processing, Volume 9, Issue 1, Jan. 2000, pp. 80 – 87.

[7] Li-hong Zhao, Xiao-Lin Sun, Ji-Hing Liu, Xin-He Xu, "Face Detection Based On Skin Color," Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, August 2004.

[8] H.Wang and S-F Chang, "A highly efficient system for automatic face region detection in MPEG video," IEEE

Trans. Circuits Syst. Video Tech. vol.7 no.4, pp. 615 – 628, 1997.
[9] Min Jiang, GuiMin He, ZhaoHui Gan, "Extending active shape models with color information for facial features localization," IEEE Int. Workshop VLSI Design & Video Tech. Suzhou, China, May 2005.
[10] Yusuke Nara, Jianming Yang, Yoshikazu Suematsu, "Face Detection Using the Shape of Face with Both Color and Edge," Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, Singapore, December, 2004.
[11] Paul Viola, Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, Dec. 2001.
[12] Paul Viola, Michael Jones, "Robust Real Time Object Detection," IEEE ICCV Workshop Statistical and Computational Theories of Vision, July 2001.
[13] Taigun Lee, Sung-Kee Park, Mignon Park, "A New Facial Features and Face Detection Method for Human-Robot Interaction," Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, Spain, April 2005.
[14] El Sayed M.Saad, Mohiy M.Hadhoud Moawad I.Moawad, Mohamed El Halawany, Alaa M. Abbas, "Detection of faces in a color natural scene using skin color classification and template matching," 22th National Radio Science Conference March 15-17, 2005, Cairo, Egypt.
[15] Bardia Mohabbati, Shohrch Kasaci, "An Efficient Wavelet/Neural Networks-Based Face Detection Algorithm," IEEE, 2005.
[16] Peng Wang, Qiang Ji, "Learning Discriminant Features for Multi-View Face and Eye Detection," Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Reconnition (CVPR 2005).
[17] Peichung Shih and Chengjun Liu,"Face Detection Using Distribution-based Distance and Support Vector Machine," Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2005).
[18] Son Lam Phung, Abdesselam Bouzerdoum and Douglas Chai, "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison," IEEE Transactions on Pattern Analysis And Machine Intellegence, Vol. 27, No. 1, January 2005.
[19] Chang Huang, Haizhou AI1, Yuan LI1and Shihong Lao, "Vector Boosting for Rotation Invariant Multi-View Face Detection," Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05).
[20] the CVL face database, http://lrv.fri.uni-lj.si/index.html