ICIC Express Letters
Part B: Applications
Volume **7**, Number **4**, April **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 781–787

AUTOMATIC DETECTION OF HUMAN FACES IN COLOR IMAGES VIA CONVOLUTIONAL NEURAL NETWORKS

Haitao Sang[1,2,*] and Zhen Zhou[1]

[1]College of Measurement and Communication Engineering
Harbin University of Science and Technology
No. 52, Xuefu Road, Nangang District, Harbin 150080, P. R. China
*Corresponding author: sanghaitao79@163.com

[2]College of Computer and Information Engineering
Heilongjiang University of Science and Technology
No. 2468, Puyuan Road, Songbei District, Harbin 150022, P. R. China

Received September 2015; accepted December 2015

Abstract. *In order to detect human face efficiently under robust conditions like complex background, different face positions and blurred image, we presented a method to detect human faces in color images via convolutional neural networks. We take skin color as a tool for detection in the YCbCr space, and give an exact segmentation of human faces with different sizes and orientations. Then using the image of face color segmentation as the input to convolutional neural networks (CNN), which is designed to provide good training and detection under translation, scaling and other transformations of the input, we realize an effective faces detection. Finally, a comparative study of different face detections is tested and analyzed, which proves the effectiveness of our method in such as computation time, correctness and robustness.*
Keywords: Human faces, Convolutional neural networks, Detection, YCbCr space

1. **Introduction.** Face detection is considered to be a key requirement in many applications such as biometrics, facial recognition systems, video surveillance, and human computer interface. Therefore, reliable face detection is required for the success of these applications. The task of human facial feature extraction is not easy, because human face varies from person to person. The race, gender, age and other physical characteristics of an individual have to be considered thereby creating a challenge in computer vision.

In recent years, there are many methods to be developed to make face detection. Initially, Viola-Jones face detection [1] uses Haar-like features for image classification. The crude quality of the features allows for an extremely fast classifier for detecting faces, yet also produces a mildly inaccurate one. Using color spaces to get a new skin color to detect face, Singh et al. [2] have combined RGB, YCbCr and HSI color spaces to get a new skin color based on face detection algorithm. Similarly, Aldasouqi et al. [3] propose a fast algorithm for detecting faces using morphology-based techniques in HSV color space. However, to achieve a high recognition accuracy, we still need to find a more effective method. In order to accelerate the detection speed, Hu and Li [4] present an algorithm to detect human face which is based on principle component analysis (PCA). Combining the ratio of width and length, they get the accurate location of face. However, they cannot deal with the image containing many persons very well. Then Tarun et al. [5] give the novel face detection approach relying on artificial neural network (ANN), which can be used to detect faces by using fast Fourier transformation (FFT). Later using fusion of PCA and artificial neural network, Shende and Patel [7] present one framework for efficient face detection. However, the detection rate above is not very high. Cheung [6] trains a convolutional neural network to distinguish between images of human faces, but he does not consider and test the blurred and noise image. Recently, Ban et al. [8] propose a face detection method based on skin color likelihood via a boosting algorithm which emphasizes skin color information while deemphasizing non-skin color information. The proposed method shows good tolerance to face in complex background. In order to increase the face detection rate when rotation and tilt of human face, Zheng and Yao [9] propose a new method called DP-AdaBoost algorithm to detect multi-angle human face and improve the correct detection rate. Compared with the classical AdaBoost algorithm [10], the textual DP-AdaBoost (Differential Projection AdaBoost) algorithm can reduce false rate significantly and improve hit rate in multi-angle face detection. However, a drawback of the proposed techniques above is that they fail to detect faces exactly when the image is blurred or pose variation, meanwhile keeping a fast speed of detection. We are motivated by the disadvantages, and seek to detect human faces in color images with the help of face color segmentation and convolutional neural networks. Using face color segmentation, we can obtain exact segmentation of human faces with different sizes and orientations, which can help us to detect face further via convolutional neural networks.

The rest of this paper is organized as follows. The face color segmentation is given in Section 2. Then convolutional neural networks is designed in Section 3. Finally, a numerical example is discussed in Section 4, and conclusions are drawn in Section 5.

2. **Face Color Segmentation.**

Color is a prominent feature of human faces. Using skin color as a primitive feature for detecting face regions has several advantages. In particular, processing color is much faster than processing other facial features. Furthermore, color information is invariant to face orientations. However, even under a fixed ambient lighting, people have different skin color appearance. In order to effectively exploit skin color for face detection, we need to find a feature space, in which human skin colors cluster tightly together and reside remotely to background colors. The YCbCr space is perceptually uniform, and it separates luminance and chrominance presenting compactness of the skin distribution cluster. Human skin forms a relatively tight cluster in color space even when different races are considered, hence learning the probability of skin color through the chromaticity information of YCbCr space could be helpful.

We adopt the YCbCr color space to give face color segmentation. Many studies found that the chrominance components of the skin-tone color are independent of the luminance component. Hence, in our implementation, only the CbCr components (the chrominance components) are used to model the distribution of skin colors.

The YCbCr space can be easily obtained from the RGB space, and the visual differences between them are shown in Figure 1. Equation (1) shows the actual conversion from RGB

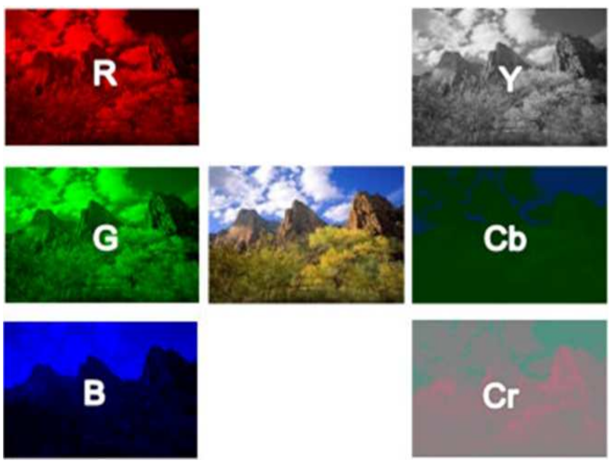

FIGURE 1. The visual differences between RGB space and YCbCr space

space to YCbCr space by a simple matrix operation.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \tag{1}$$

In the CbCr subspace, the distribution of skin colors is modeled with a multivariate Gaussian mixture model (GMM); thereby the parameters of the Gaussian mixture model are estimated using the standard Expectation-Maximization (EM) algorithm [11]. That is, each skin color value is viewed as a realization from a Gaussian mixture model S consisting of Gaussian components $S_{i\,i=1}^{\,k}$, each characterized by its mean vector μ_{S_i} and covariance matrix Σ_{S_i}, in some proportions $\gamma_1, \gamma_2, \ldots, \gamma_k$, where $\sum_{i=1}^{k} \gamma_i = 1$ and $\gamma_i > 0$; the number of Gaussian components k is 5 in our implementation. The probability that a pixel j with color value X_j belongs to the skin color model S can be computed as

$$\begin{aligned} p(X_j \mid S) &= \sum_{i=1}^{k} \gamma_i p(X_j \mid S_i) \\ &= \sum_{i=1}^{k} \frac{\gamma_i}{(2\pi)^{3/2} \mid \Sigma_{S_i} \mid^{1/2}} \exp\left\{ -\frac{1}{2}(X_j - \mu_{S_i})^T \sum_{S_i}^{-1} (X_j - \mu_{S_i}) \right\} \end{aligned} \tag{2}$$

Similarly, the pixels of the background scene are also modeled with a GMM. To account for the variety of background colors, the number of Gaussian components in the background GMM is larger than that of the skin color. In our implementation, the number of Gaussian components is set to 8 for the background GMM. Let B_i denote the background model consisting of a mixture of Gaussian components $B_{i\,i=1}^{\,k}$, each characterized by its mean vector μ_{B_i} and covariance matrix Σ_{B_i}, in some proportions $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $\sum_{i=1}^{k} \alpha_i = 1$ and $\alpha_i > 0$. The probability that a pixel j with color value X_j belongs to the background model B can be computed as

$$\begin{aligned} p(X_j \mid B) &= \sum_{i=1}^{k} \alpha_i p(X_j \mid B_i) \\ &= \sum_{i=1}^{k} \frac{\alpha_i}{(2\pi)^{3/2} \mid \Sigma_{B_i} \mid^{1/2}} \exp\left\{ -\frac{1}{2}(X_j - \mu_{B_i})^T \sum_{B_i}^{-1} (X_j - \mu_{B_i}) \right\} \end{aligned} \tag{3}$$

After obtaining the GMMs of skin colors and background colors, the segmentation of human faces can then be done by maximum likelihood classification of pixels within a test image. Specifically, given the background model B and the skin model S, a pixel with color value X is considered to be skin pixel if

$$p(X \mid S) > p(X \mid B) \tag{4}$$

3. **Design of Convolutional Neural Networks.** When an artificial neural network has learned to recognize a certain feature at one place in the input image, it needs to relearn to recognize that same feature at other positions in the input image. CNN has convolutional layers to tackle this problem. The convolutional layers are designed to provide invariance under translation, scaling and other transformations of the input. Figure 2 shows a model of a CNN. The number in Figure 2 refers to image sizes and dimensions.

3.1. **Convolution layers.** At the convolution layer, the previous layer's feature maps are convolved with learnable kernels and put through the activation function to form the output feature map. Each output map may combine convolutions with multiple input

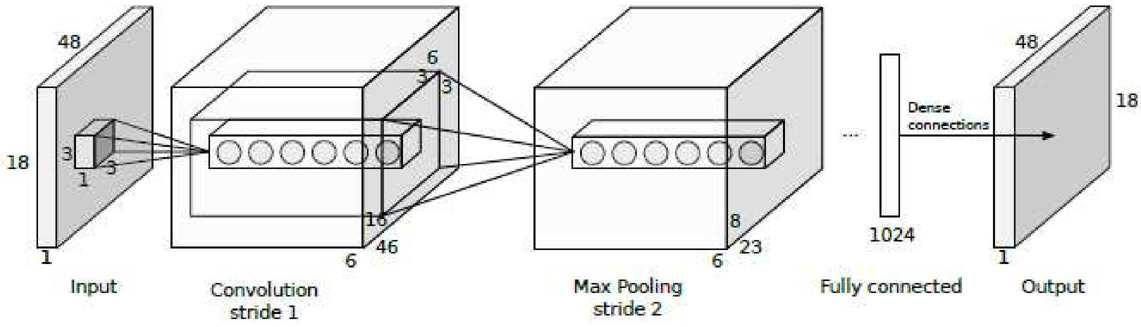

Figure 2. CNN model

maps. In general, we have that

$$X_j^\ell = f\left(\sum_{i \in M_j} X_i^{\ell-1} * K_{i,j}^\ell + b_j^\ell\right) \tag{5}$$

where ℓ denotes the current layer. M_j represents a selection of input maps, and the convolution is of the "valid" border handling type when implemented in MATLAB. K is kernel function. Each output map is given an additive bias b; however, for a particular output map, the input maps will be convolved with distinct kernels. That is to say, if output map j and map k both sum over input map i, then the kernels applied to map i are different for output maps j and k.

3.2. **Sub-sampling layers.** A sub-sampling layer produces downsampled versions of the input maps. If there are N input maps, then there will be exactly N output maps, although the output maps will be smaller. More formally,

$$X_j^\ell = f\left(\beta_j^\ell down\left(X_j^{\ell-1}\right) + b_j^\ell\right) \tag{6}$$

where $down()$ represents a sub-sampling function. Typically this function will sum over each distinct n-by-n block in the input image so that the output image is n-times smaller along both spatial dimensions. Each output map is given its own multiplicative bias β and an additive bias b.

3.3. **Learning combinations of feature maps.** In general, the input maps that are combined to form a given output map are typically chosen by hand. We can, however, attempt to learn such combinations during training. Let α_{ij} denote the weight given to input map i when forming output map j. Then output map j is given by

$$X_j^\ell = f\left(\sum_{i=1}^{N_{in}} \alpha_{i,j}\left(X_i^{\ell-1} * K_i^\ell\right) + b_j^\ell\right) \tag{7}$$

subject to the constraints

$$\sum_i \alpha_{ij} = 1, \text{ and } 0 \le \alpha_{ij} \le 1 \tag{8}$$

These constraints can be enforced by setting the α_{ij} variables equal to the softmax over a set of unconstrained, underlying weights c_{ij}:

$$\alpha_{ij} = \frac{\exp(c_{ij})}{\Sigma_K \exp(c_{ij})} \tag{9}$$

Because each set of weights c_{ij} for fixed j is independent of all other such sets for any other j, we can consider the updates for a single map and drop the subscript j. Each map is updated in the same way, except with different j indices.

The derivative of the softmax function is given by

$$\frac{\partial \alpha_k}{\partial c_i} = \delta_{ki}\alpha_i - \alpha_i\alpha_k \tag{10}$$

(where here δ is used as the Kronecker delta), while the derivative of the error function E with respect to the α_i variables at layer ℓ is

$$\frac{\partial E}{\partial \alpha_i} = \frac{\partial E}{\partial \mu^\ell}\frac{\partial \mu^\ell}{\partial \alpha_i} = \sum_{u,v}\left(\delta^\ell \circ \left(X_i^{\ell-1} * K_i^\ell\right)\right)_{uv} \tag{11}$$

Here, δ^ℓ is the sensitivity map corresponding to an output map with inputs u. Again, the convolution is the "valid" type so that the result will match the size of the sensitivity map. We can now use the chain rule to compute the gradients of the error function with respect to the underlying weights c_i:

$$\frac{\partial E}{\partial c_i} = \sum_k \frac{\partial E}{\partial \alpha_k}\frac{\partial \alpha_k}{\partial c_i} = \alpha_i\left(\frac{\partial E}{\partial \alpha_i} - \sum_k \frac{\partial E}{\partial \alpha_k}\alpha_k\right) \tag{12}$$

4. **A Numerical Example.** The method is tested on many different types of people and faces in MATLAB2014a. We have 10 filters of dimension 3×3, input image size 512×512 and a contiguous 2×2 pooling region. Quantitative and qualitative analyses of the results are provided in this section. The following aspects are considered: coding practice, computation time, correctness and robustness. Overall, our method is considered as correct and effective.

A test image shown in Figure 3 is taken and the parts of the image having colors under the range of skin colors are highlighted. This image contains human faces with different sizes and orientations, which imposes great challenge on our face detector.

Figure 3. Test image of human faces with different sizes and orientations

This image in Figure 3 is studied as a test image and the image is converted into a binary image in which the skin colors are highlighted as white and the rest as black. The output of this filtering process in the YCbCr space is shown in Figure 4. The faces outlines of these people have clear segmentation.

Then using face color segmentation and CNN, two faces can be also exactly detected, and those faces are outputted (the output is a bounding box roughly around the human faces) in Figure 5(a). In order to prove the correctness and robustness of face detection via our method, we blur the image and give the detection result of the blurred image in Figure 5(b).

From Figure 5, we can see that we can obtain exact detection result even the image is blurred, which shows that our method is robust and effective.

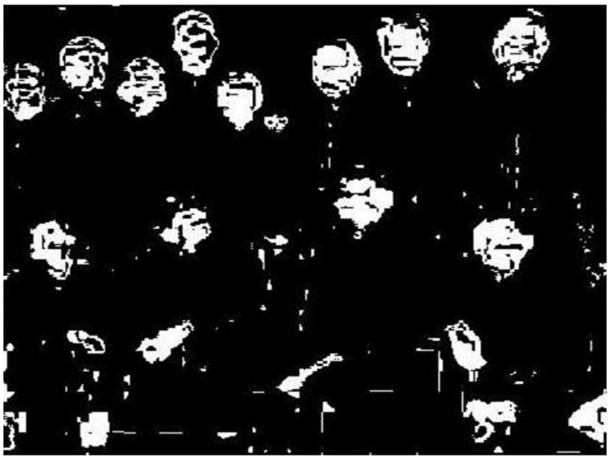

Figure 4. The image after passing through YCbCr filter

(a) Face detection for initial image (b) Face detection for blurred image

Figure 5. Face detection using our method

To give an objective evaluation, we use the CMU database of face images [12] and test the detection effectiveness on eight images which contain fifty persons in total. We count the missed faces, the hitting faces, detection rate and computation time in the testing images for comparison with the traditional methods such as PCA [4] and ANN [5] in Table 1.

Table 1. Overall performance of the face detector via different methods

Methods	**Missed faces**	**Hitting faces**	**Detection rate**	**Time (seconds)**
PCA [4]	8	42	84%	30.1
ANN [5]	5	45	90%	57.5
Our method	2	48	96%	33.3

From Table 1, we can see that our method could reduce false rate, improve hitting rate in multi-character face detection, and obtain less computation time. So our method performs better effect of face detection over PCA and ANN.

5. **Conclusions.** In this paper, an efficient method of face detection has been proposed and implemented to improve detection effectiveness. Our method not only can detect edge in images precisely but also has a good robustness. Experimental results show that we can acquire a better detection effect even the image is containing many people or blurred. Also we give an objective evaluation in the missed faces, the hitting faces, detection rate

and computation time for comparison with the traditional methods such as PCA and ANN. The result shows that our method performs better effect of face detection.

Although the proposed approach has achieved a better detection effect, and the detection time is still relatively long, how to accelerate the detection speed is an interesting research direction which we are working on.

Acknowledgment. This work was supported by the Science and Technology Foundation of Heilongjiang Province Education Department of China (No. 12541710).

REFERENCES

[1] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, *Proc. of Conference on Computer Vision and Pattern Recognition*, pp.511-518, 2001.

[2] S. K. Singh, D. S. Chauhan, M. Vatsa and R. Singh, A robust skin color based face detection algorithm, *Tamkang Journal of Science and Engineering*, vol.6, no.4, pp.227-234, 2003.

[3] I. Aldasouqi, M. Hassan and M. Hassan, Smart human face detection system, *International Journal of Computers*, 2011.

[4] Y. Hu and G. Li, Face detection based on PCA in color images, *Microcomputer Information*, vol.24, no.6, pp.64-66, 2008.

[5] K. Tarun, K. V. Singh and M. Shekhar, Artificial neural network in face detection, *International Journal of Computer Applications*, vol.14, no.3, pp.5-7, 2011.

[6] B. Cheung, Convolutional neural networks applied to human face classification, *IEEE the 11th International Conference on Machine Learning and Applications*, pp.580-583, 2012.

[7] S. Shende and R. Patel, Efficient face detection using PCA and ANN techniques, *International Journal on Advanced Computer Theory and Engineering*, vol.2, no.5, pp.155-159, 2013.

[8] Y. Ban, S. K. Kim, S. Kim, K. A. Toh and S. Lee, Face detection based on skin color likelihood, *Pattern Recognition*, vol.47, no.4, pp.1573-1585, 2014.

[9] Y. Y. Zheng and J. Yao, Multi-angle face detection based on DP-AdaBoost, *International Journal of Automation and Computing*, vol.12, no.4, pp.421-431, 2015.

[10] K. E. Li and L. P. Wen, Face detection based on modified AdaBoost algorithm, *Opto-Electronic Engineering*, vol.39, no.1, pp.113-118, 2012.

[11] F. Dellaert, The expectation maximization algorithm, *Georgia Institute of Technology*, vol.3, no.5, pp.1397-1400, 2002.

[12] *CMU Database of Face Images*, http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html.