Face Detection in Color Images Using Skin Segmentation

Mohammadreza Hajiarbabi, Arvin Agah

Abstract:
Face detection which is a challenging problem in computer vision, can be used as a major step in face recognition. The challenges of face detection in color images include illumination differences, various cameras characteristics, different ethnicities, and other distinctions. In order to detect faces in color images, skin detection can be applied to the image. Numerous methods have been utilized for human skin color detection, including Gaussian model, rule-based methods, and artificial neural networks. In this paper, we present a novel neural network-based technique for skin detection, introducing a skin segmentation process for finding the faces in color images.

Keywords: skin detection, neural networks, face detection, skin segmentation, and image processing

1. Introduction

Face recogni**tion is an active area of research in image processing** and computer vision. Face recognition in color images consists of three main phases. First is skin detection, in which the human skin is detected in the image. Second is face detection in which the skin components found in the first phase are determined to be part of human face or not. The third phase is to recognize the detected faces. This paper focuses on skin detection and face detection phases.

Face detection is an important step not only in face recognition systems, but also in many other computer vision systems, such as video surveillance, human-computer interaction (HCI), and face image retrieval systems. Face detection is the initial step in any of these systems. The main challenges in face detection are face pose and scale, face orientation, facial expression, ethnicity and skin color. Other challenges such as occlusion, complex backgrounds, inconsistent illumination conditions, and quality of the image further complicate face detection in images. The skin color detection is also an important part in many computer vision applications such as gesture recognition, hand tracking, and others. Thus, skin detection is also challenging due to different illumination between images, dissimilar cameras and lenses characteristics, and the ranges of human skin colors due to ethnicity. One important issue in this field is the pixels' color values which are common between human skin and other entities such as soil and other common items [2].

There are a number of color spaces that can be used for skin detection. The most common color spaces are RGB, YCbCr, and HSV. Each of these color spaces has its own characteristics.

The RGB color space consists of red, green and blue colors from which other colors can be generated. Although this model is simple, it is not suitable for all applications [16]. In the YCbCr color space, Y is the illumination (Luma component), and Cb and Cr are the Chroma components. In skin detection, the Y component can be discarded because illumination can affect the skin. The equations for converting RGB to YCbCr are as follows:

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = B - Y$$
$$Cr = R - Y$$

The HSV color space has three components, namely, H (Hue), S (Saturation), and V (Value). Because V specifies the brightness information, it can be eliminated in skin detection. The equations for converting RGB to HSV are as follows:

$$R' = \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255}$$

$$Cmax = max\,(R', G', B'), Cmin = min\,(R', G', B'),$$
$$x = Cmax - Cmin$$

$$H = 60 * \left(\frac{G' - B'}{x}\ mod6 \right),\ Cmax = R'$$

$$H = 60 * \left(\frac{B' - R'}{x} + 2 \right),\ Cmax = G'$$

$$H = 60 * \left(\frac{R' - G'}{x} + 4 \right),\ Cmax = B'$$

$$S = \frac{x}{Cmax} for\ x\ not\ being\ zero\ other\ wise\ S = 0$$

$$V = Cmax$$

2. Skin Color Detection

Three skin detection methods of Gaussian, rule-based, and neural networks are discussed in this section.

2.1. Gaussian Methods

This technique which was proposed in [20], uses the Gaussian model in order to find the human skin in an image. The RGB image is transformed to the YCbCr color space. The density function for Gaussian variable $X = (Cb\ Cr)^T \in R^2$ is:

$$f(Cb\ Cr) = \frac{1}{2\pi |C|^{1/2}} exp\{-\frac{1}{2}(X-\mu)^T C^{-1}(X-\mu)\}$$

Where $X = \begin{pmatrix} Cb \\ Cr \end{pmatrix}$, $\mu = \begin{pmatrix} \mu_{Cb} \\ \mu_{Cr} \end{pmatrix}$, $C = \begin{pmatrix} C_{CbCb} & C_{CbCr} \\ C_{CrCb} & C_{CrCr} \end{pmatrix}$,

and the parameters are:

$$\mu_{skin} = \begin{pmatrix} 112.1987 \\ 151.3993 \end{pmatrix}, \quad C_{skin} = \begin{pmatrix} 89.3255 & 32.2867 \\ 32.2867 & 252.9336 \end{pmatrix}$$

The parameters are calculated using sets of training images. For each pixel value, the density function is calculated; however, only the (Cb Cr) value is used because the Y component has the illumination information which is not related to skin color. The probability value of more than a specified threshold is considered as skin. The final output is a binary image where the non-skin pixels are shown by black and human skin by white. It is worth to mention that the amount for parameters μ and C are calculated using a specified training samples and can be varied by using other training samples.

2.2. Rule-Based Methods

Skin detection based on rule-based methods has been used in several research efforts as the first step in face detection. Chen et al. analyzed the statistics of different colors [5]. They used 100 images for training, consisting of skin and non-skin in order to calculate the conditional probability density function of skin and non-skin colors.

After applying Bayesian classification, they determined the rules and constraints for the human skin color segmentation. The rules are:

$$r(i) > \alpha, \beta_1 < r(i) - g(i) < \beta_2, \gamma_1 < r(i) - b(i) < \gamma_2$$
$$\sigma_1 < (g(i) - b(i)) < \sigma_2$$

With

$$\alpha = 100, \beta_1 = 10, \beta_2 = 70, \gamma_1 = 24, \gamma_2 = 112, \sigma_1 = 0$$
$$\text{and } \sigma_2 = 70$$

Although this method works on some images perfectly, the results are not reliable on images with complex background or uneven illumination.

Kovac et al. introduced two sets of rules for images taken indoor or outdoor [11]. These rules are in RGB space, where each pixel that belongs to human skin must satisfy certain relations.
For indoor images:

$$R>95 ,G>40 ,B>20,$$
$$max \{R, G, B\} - min \{R, G, B\} > 15,$$
$$|R-G|>15 , R>G , R>B$$

For images taken in daylight illumination:

$$R>220 ,G>210 ,B>170 ,|R-G|\leq 15 ,$$
$$R>B , G>B$$

Kong et al. presented rules that use the information from both HSV and normalized RGB color spaces [10]. They suggested that although in normalized RGB the effect of intensity has been reduced, it is still sensitive to illumination. Therefore, they also use HSV for skin detection. Each pixel that satisfies these rules is considered to be a human skin pixel:

$$0.4 \leq r \leq 0.6, \quad 0.22 \leq g \leq 0.33, \quad r > g > \frac{1-r}{2},$$
$$0 \leq H \leq 0.2, \quad 0.3 \leq S \leq 0.7, \quad 0.22 \leq V \leq 0.8$$

2.3. Neural Network Methods

Neural network has been used in skin color detection in a number of research projects. Doukim et al. use YCbCr as the color space with a Multi-Layer Perceptron (MLP) neural network [6]. They used two types of combination strategies, and several rules are applied. A coarse to fine search method was used to find the number of neurons in the hidden layer. The combination of Cb/Cr and Cr features produced the best result.

Seow et al. use the RGB as the color space which is used with a three-layered neural network [14]. Then the skin regions are extracted from the planes and are interpolated in order to obtain an optimum decision boundary and the positive skin samples for the skin classifier.

Yang et al. use YCbCr color space with a back propagation neural network [21]. They take the luminance Y and sort it in ascending order, dividing the range of Y values into a number of intervals. Then the pixels whose luminance belong to the same luminance interval are collected. In the next step, the covariance and the mean of Cb and Cr are calculated and are used to train the back propagation neural network. Another example of methods of human skin color detection using neural network can be found in [1].

3. Skin Color Detection

The novel approach presented in this paper is based on skin detection using neural networks with hybrid color spaces.

Neural network is a strong tool in learning, so it was decided to use neural network for learning pixels' colors, in order to distinguish between what is face skin pixel and what is a non-face skin pixel. We decided to use information from more than one color space instead of using just the information from one color space. We gathered around 100,000 pixels for face and 200,000 for non-face pixels from images chosen from the Web.

Choosing images for the non-skin is a rather difficult task, because that is an enormous category, i.e., everything which is not human skin is non-skin. We tried to choose images from different categories, es-

pecially those which are very similar to human skin color, such as sand, surfaces of some desks, etc. We used such things in training the neural network so that the network can distinguish them from human skin.

For the implementation, a multi-layer perceptron (MLP) neural network was used. Several entities can be used as the input to the neural network, namely, RGB, HSV (in this case V is not used because it has the illumination information which is not suitable for skin detection), YCbCr (Y is also not used because it has the illumination information). The number of outputs can be one or two. If there is just one output, then a threshold can be used. For example, an output greater than 0.5 indicates that the input pixel belongs to skin, and less than that shows that it belongs to non-skin. For two outputs, one output belongs to skin and the other to non-skin. The larger value of the two outputs identifies the class of the pixel.

Around half of the samples were used for training and the rest for testing/validation. Different numbers of neurons were examined in the hidden layer, ranging from two nodes to 24 nodes. The networks which produced better results were chosen for the test images. For most of the networks, having 16 or 20 nodes in the hidden layer produced better results in comparison to other number of neurons in the hidden layer. A combination of the different color space CbCrRGBHS was used as the input. Y and V were eliminated from YCbCr and HSV because they contain illumination information.

We trained several different neural networks [7] and tested the results on the UCD database, using MATLAB (developed by MathWorks) for implementation. The UCD database contains 94 images from different ethnicities. The images vary from one person in the image to multiple people. The UCD database also contains the images after cropping the face skin. The Feed Forward neural network was used in all the experiments. We considered one node in the output. If the value of the output node is greater than 0.5, then the pixels belongs to human skin, otherwise it is not a human skin.

The experimental results are reported as precision, recall, specificity and accuracy.

Precision or positive predictive value (PPV):
$$PPV = TP/((TP+FP))$$

Sensitivity or true positive rate (TPR) equivalent with hit rate, recall:

$$TPR = TP/P = TP/((TP+FN))$$

Specificity (SPC) or true negative rate:

$$SPC = TN/N = TN/((FP+TN))$$

Accuracy (ACC):

$$ACC = ((TP+TN))/((P+N))$$

In the skin detection experiments, P is the number of the skin pixels; N is the number of the non-skin pix-els. TP is the number of the skin pixels correctly classified as skin pixels. TN is the number of the non-skin pixels correctly classified as non-skin pixels. FP is the number of the non-skin pixels incorrectly classified as skin pixels. FN is the number of the skin pixels incorrectly classified as non-skin pixels.

4. Experimental Results

We generated a vector consisting of the information of the color spaces CbCrRGBHS and yielded the results in Table 1.

Another neural network was designed with having the same input but different nodes in the output. In this experiment two nodes were chosen for the output, one for the skin and the other for the non-skin (higher value determines class).

The results for CbCrRGBHS vector are listed in Table 2. The results show that in case of recall and precision we have some improvement, but the precision has decreased.

Table 3 shows the result of other methods discussed compared to our best results on using the UCD database. Comparing the other methods with the result we have from the CbCrRGBHS vector shows that our result is better in precision, specificity and accuracy. Our method [7] accepts fewer non-skin pixels as skin comparing to other methods.

It should be noted that there is a tradeoff between precision and recall. If we want to have high recall (recognizing more skin pixels correctly) then it is highly possible to recognize many non-skin pixels as human skin which will reduce the precision and vice versa.

Figures 1 to 7 illustrate some of our experimental results on images from the UCD database. These are produced using the CbCrRGBHS vector and two outputs for the neural network. The second image is the output from the neural network and the third image is after applying morphological operation. We first filled the holes that were in the image. After that, we applied erosion, followed by dialation operation [7]. The structuring element which was used by us was 3*3. This size had better results than other structuring elements.

5. Methods for Face Detection

After the skin detection phase, the next step is to use a face detection algorithm to detect the faces in the image. Several methods have been used for face detection. In this section we discuss the common methods which have been used in this field, namely, Rowley *et al.* [13] and Viola, Jones [19].

5.1. Rowley Method for Face Detection

Rowley et al. used neural networks, detecting upright frontal faces in gray-scale images [13]. One or more neural networks are applied to portions of an image and absence or presence of a face is decided. They used a bootstrap method for the training, which means that they add the images to the training set as the training progresses. Their approach has two stages. First a set of neural network-based filters are

	Precision	Recall	Specificity	Accuracy
CbCrRGBHS	77.73	41.35	95.92	81.93

Table 1. Accuracy results for CbCrRGBHS

	Precision	Recall	Specificity	Accuracy
CbCrRGBHS	71.30	50.25	93.43	82.36

Table 2. Accuracy results for CbCrRGBHS

	Precision	Recall	Specificity	Accuracy
Gaussian	54.96	66.82	81.12	77.46
Chen	63.75	51.13	89.98	80.02
Kovac	62.51	69.09	85.71	81.45
Kong	37.47	14.58	91.61	71.87
CbCrRGBHS	71.30	50.25	93.43	82.36

Table 3. Accuracy results for other methods

applied to an image. These filters look for faces in the image at different scales. The filter is a 20*20 pixel region of the image and the output of the image is 1 or -1, where 1 indicates that the region belongs to face and -1 indicates that the region contains no face. This filter moves pixel by pixel through the entire image. To solve the problem for faces bigger than this size, the image is subsampled by a factor of 1.2.

They use a preprocessing method [17], where first the intensity values across the window are equalized. A linear function is fitted to the intensity values in the window and then it is subtracted, which corrects some extreme lightening conditions. Then histogram equalization is applied, in order to correct the camera gain and also to improve the contrast. Also an oval mask is used to ignore the background pixels.

The window is then passed to a neural network. Three types of hidden units are used. Four units which looked at 10*10 sub regions, 16 which looked at 5*5 sub regions, and 6 which looked at overlapping 20*5 horizontal stripes. These regions are chosen in order to detect different local features of the face. For example the horizontal stripes were chosen to detect eyes, eyebrows, etc.

Around 1050 face images are used for training. Images (black and white) are chosen from Web and some popular databases. For the non-face images another method is used. The reason is the face detection is quite different from other problems. The set of non-face images is much larger than face images. The steps of their method are:
– An initial set consisting of 1000 random images are generated. The preprocessing method is applied to these images.

– A network is trained using the face and non-face images.
The network is tested on an image that contained no face. The misclassified sub images is chosen (those which were considered as faces wrongly).
– 250 of these sub images are chosen randomly, the preprocessing methods are applied to them and these sub images are added into the training set as negative examples. The process is continued from the second step.

Other new ideas were utilized by [13]. In the areas which contain face, there are lots of detection because of the moving pixel by pixel of the window over the image. Also several detections are required because of using different scales over the image. They used a threshold and counted the number of the detections in each location. If the number of detections is above a specified threshold, then that location is considered as a face otherwise rejected. Other nearby detections that overlap the location classified as face are considered as error, because two faces cannot overlap. This method is called overlap elimination.

Another method that is used to reduce the number of false positives is using several networks and arbitration method between them to produce the final decision. Each network is trained with different initial weights, different random sets of non-face images, and different permutations of the images that are presented to the network. Although the networks had very close detection and error rates, the errors were different from each other. They use a combination of the networks using AND, OR and voting methods.

5.2. Viola Method for Face Detection

Viola *et al.* trained a classifier using a large number of features [19]. A set of large weak classifiers are used, and these weak classifiers implemented a threshold function on the features. They use three types of Haar features. In two-rectangle feature, the value is the subtraction between the sums of the pixels within two regions. In three-rectangle feature, the value is the sum within two outside rectangles subtracted from the inside rectangle. In four-rectangle feature, the value is the difference between diagonals of the rectangle. They use the integral image that allowed the features to be computed very fast.

Their AdaBoost learning algorithm is as follows:
– The first step is initializing the weights

$$w_{1,i} = \frac{1}{2m}, \quad \frac{1}{2l} \; for \; y_i = 0,1$$

Where m is the number of positive examples and l is the number of negative examples. $y_i = 0$ for negative examples and $y_i = 1$ for positive examples.

– For t = 1,…,T

  – Normalize the weights using

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

  Now the value of the weights will be between 0 and 1 and so is a probability distribution.
– For each feature *j* a classifier h_j is trained and uses just a single feature. The error is:

$$\epsilon_j = \sum_i \omega_i \left| h_j(x_i) - y_i \right|$$

  – The classifier h_t with the lowest error ε_t is chosen and the weights are updated using

$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$$

If example x_i is classified correctly then $e_i = 0$, otherwise $e_i = 1$ and

$$\beta_t = \frac{\epsilon_t}{1 - \varepsilon_t}$$

– The final classifier is:

$$h(x) = \{ \begin{smallmatrix} 1 & \sum_{t=1}^{T} \alpha_t \, h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{smallmatrix} \; and \; \alpha_t = \log \frac{1}{\beta_t}$$

Viola *et al.* made another contribution which was constructing a cascade of classifiers which is designed to reduce the time of finding faces in an image [19]. The beginning cascades reject most of the images, images which pass the first cascade will go to the second one, and this process continues till to the end cascade of classifiers.

Similar to the Rowley method, the Viola method includes a window that is moving on the image and decides if that window contains a face. However, Viola showed that their method is faster than Rowley [19].

5.3. Other Methods

There are some other methods which have been used for face detection. Principal component analysis (PCA) method which generates Eigen faces has been used in some approaches for detecting faces [3]. Other types of neural networks have also been used in [12] and [22]. Shavers used Support Vector Machines (SVM) for face detection [15]. Jeng used geometrical facial features for face detection [9]. They have shown that their method works for detecting faces in different poses. Hjelmas has a survey in face detection methods from single edge based
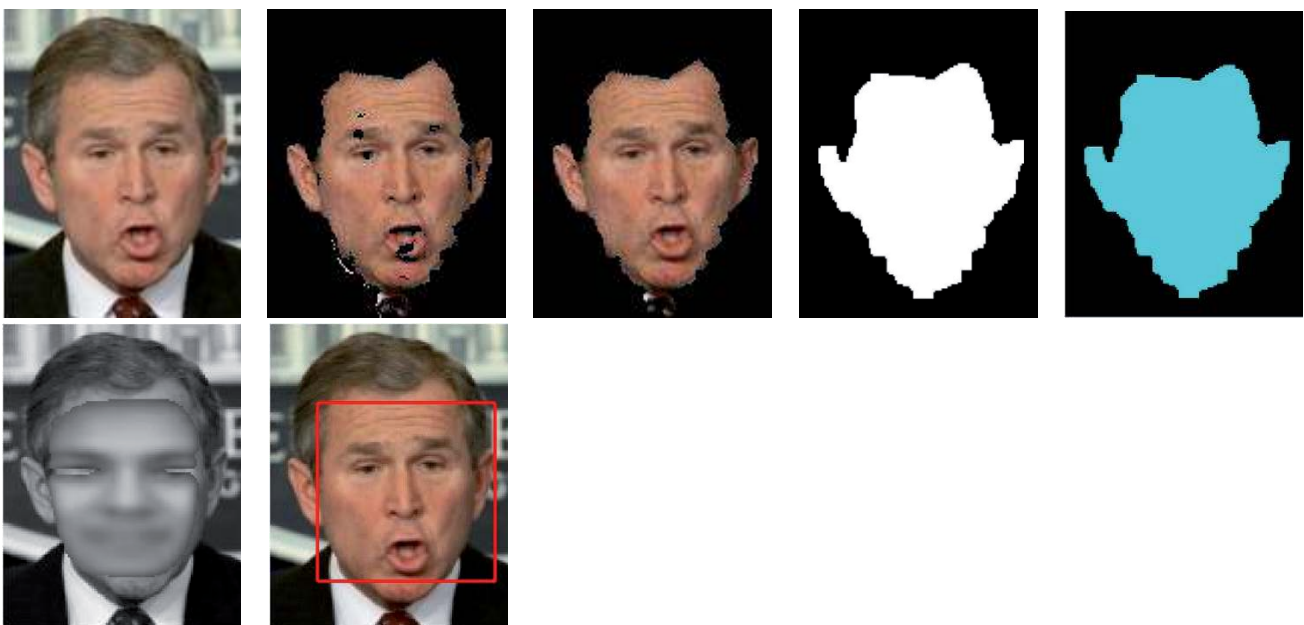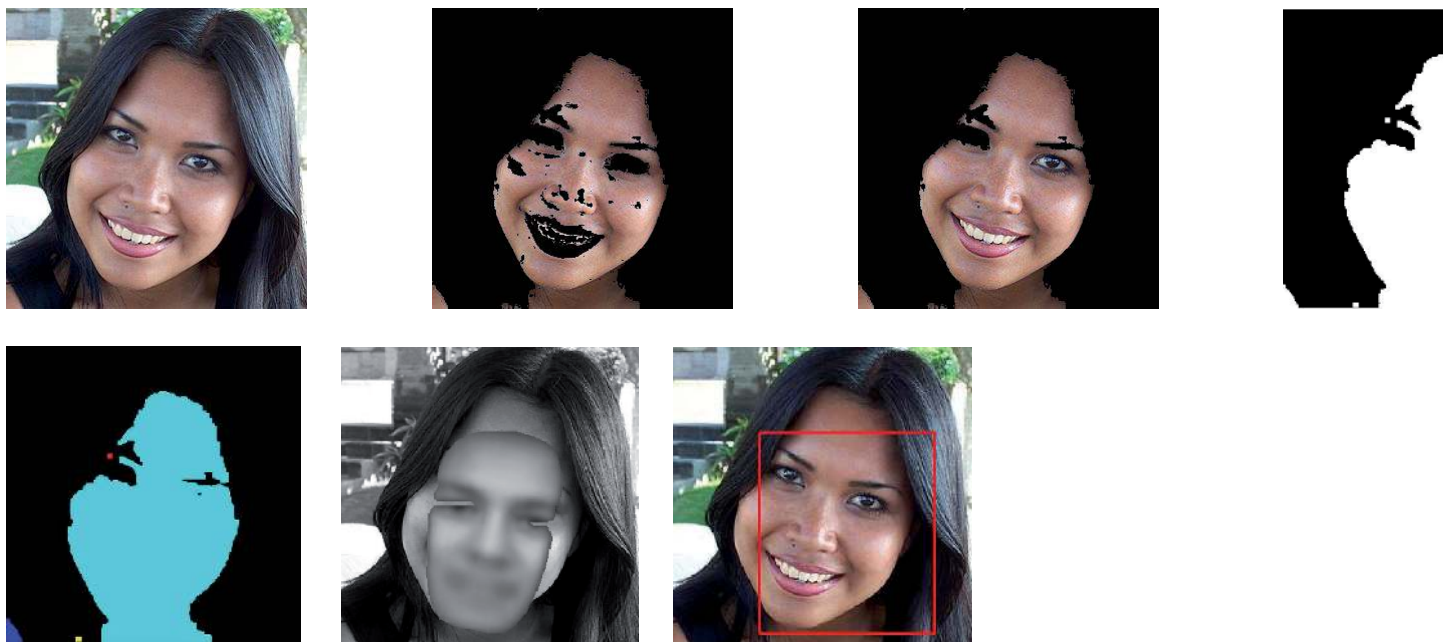

Figure 1. Experimental results
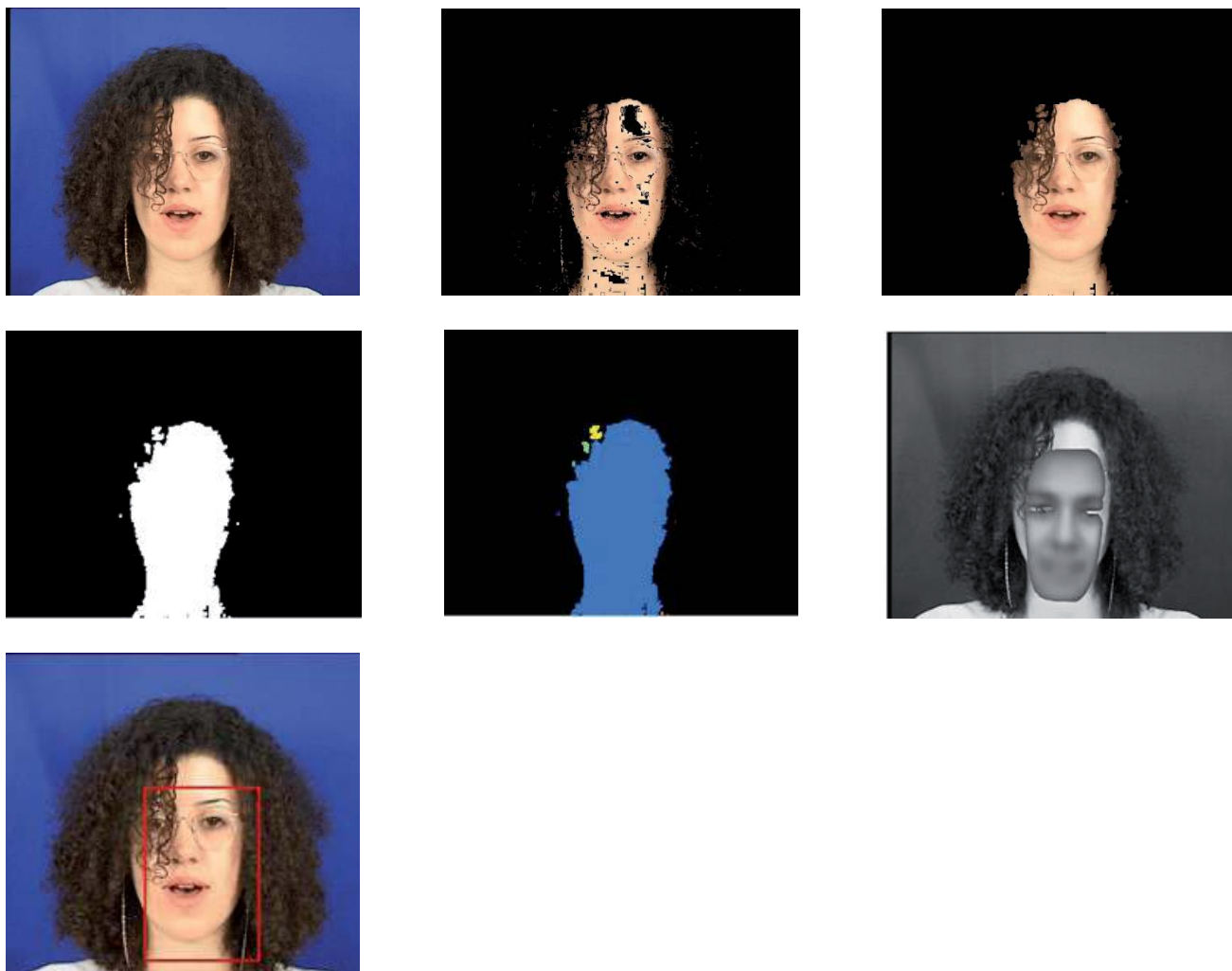
Figure 2. Experimental results

Figure 3. Experimental results

Figure 4. Experimental results

Figure 5. Experimental results, two faces with complex background
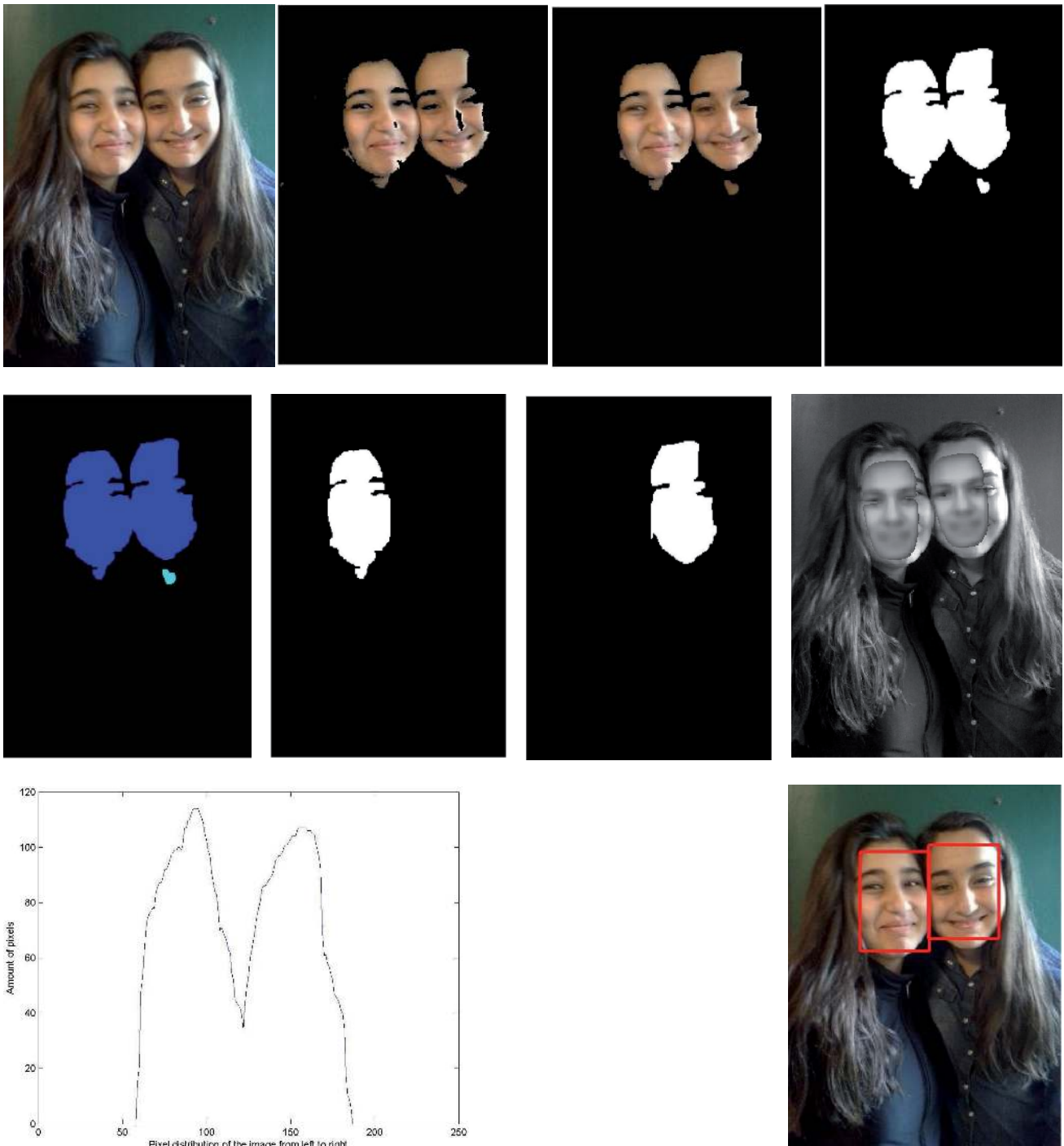
Figure 6. Experimental results, when two faces are close to each other

algorithms to high level approaches [8]. Yang also has published another survey in face detection and numerous techniques have been described in it [23].

6. Face Detection Research Approach

Rowley and Viola methods both search all areas of image in order to find faces; however, in our approach, we first divide the image into two parts, the parts than contain human skin and the other parts. After this step the search for finding human face would be restricted to those areas that just contain human skins. Therefore face detection using color images can be faster than other approaches. As mentioned in [8]

due to lack of standardized test, there is not a comprehensive comparative evaluation between different methods, and in case of color images the problem is much more because there are not many databases with this characteristic. Because of this problem it is not easy to compare different methods like Viola and Rowley methods with color based methods. But face detection using color based is faster than other methods because unlike Viola and Rowley method it does not need a window to be moved pixel by pixel on the whole image. Other papers such as [4] has also mentioned that color based methods is faster comparing to other methods.
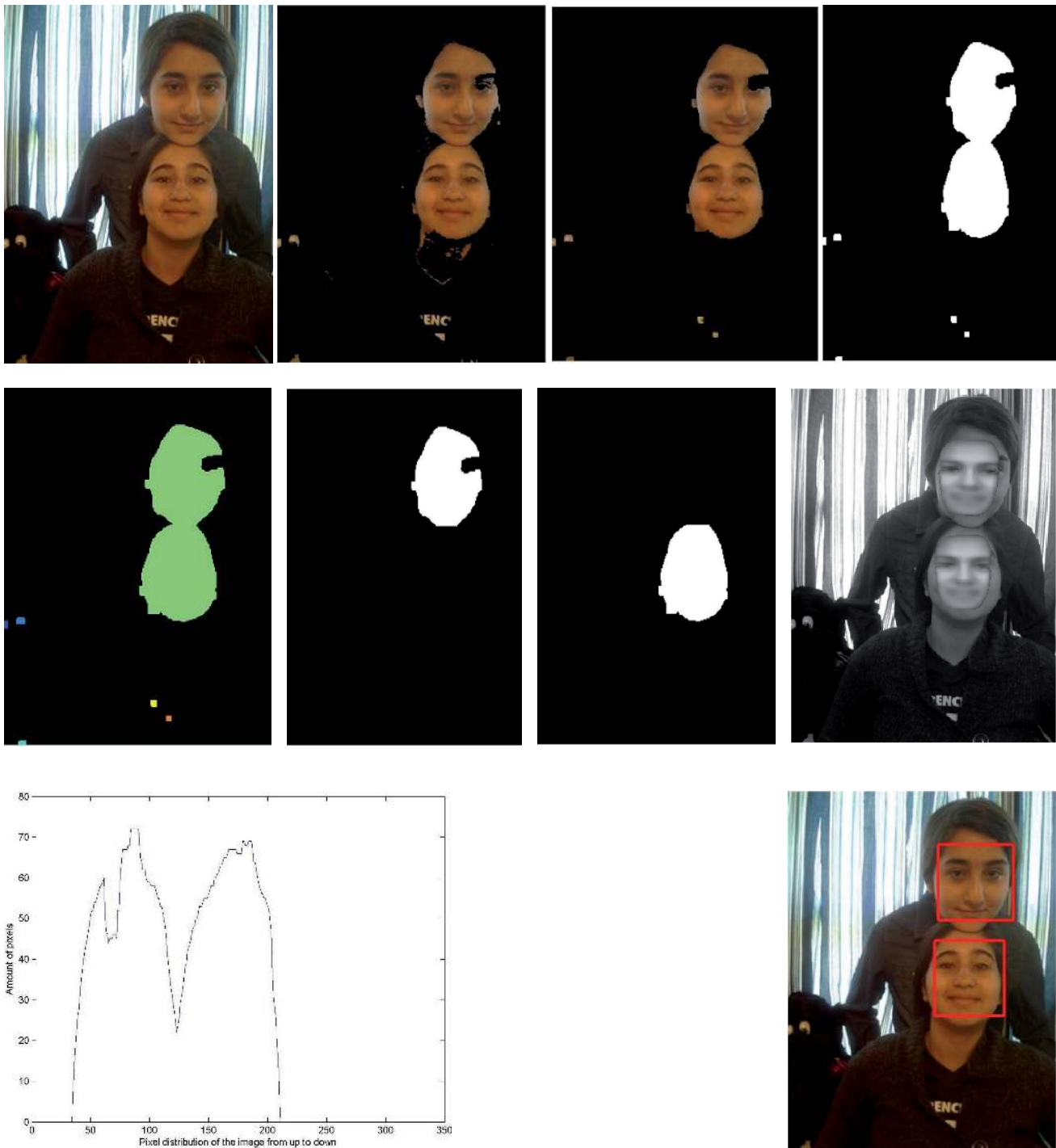
Figure 7. Experimental results, when two faces are close to each other

In color images, we use the idea that we can separate the skin pixels from the other part of the image and by using some information we can recognize the face from other parts of the body. The first step in face detection is region labeling. In this case the binary image, instead of having values 0 or 1, will have value of 0 for the non-skin part and values of 1, 2... for the skin segments which was found in the previous step [4].

The next step that can be used is the Euler test [4]. Because there are some parts in image like the eyes, eyebrows, etc. that their colors differ from the skin. By using Euler test one can distinguish face components from other components such as the hands and

arms. The Euler test counts the number of holes in each component. One main problem in Euler test is that there may be some face components which has no holes in them and also some components belonging to hands or other parts of the body with holes in them. So Euler test cannot be a reliable method and we did not use it.

At the next step, the cross correlation between a template face and grayscale image of the original image is calculated. The height, width, orientation and the centroid of each component are calculated. The template face is also resized and rotated. The center of the template is then placed on the center of the

component. The cross correlation between these two region is then calculated. If that is above a specified threshold, the component would be considered as a face region; otherwise it will be rejected [4].

We have modified this algorithm. The first revision is that we discard the components where the proportion of the height to the width was larger than a threshold, except for some of these components which will be discussed later. In this case we were sure that no arms or legs would be considered as face. Second, the lower one fourth part of image is less probable to have faces and so we set a higher threshold for that part, as that part of the image most likely belongs to feet and hands. Increasing the threshold for the lower one forth part decreased the false positive of that part of the images.

Third, for the components which are rejected we used a window consisting of the upper part of the face. We move the window across each bit of the components and calculate the correlation between the window and the area with the same size of the window. If the correlation is above certain threshold that part is considered to be face. For covering different sizes, we down sample the image (size*0.9) seven times. In this case, there may be some parts with overlapping rectangles. The rectangles around the face which had more than a specified area in common with each other are deleted and just one of them is kept.

This novel method is useful for those components where the skin detection part has not distinguished between the skin pixels and the other pixels correctly. For example, in some images some pixels from the background are also considered as skin pixels, in this case these components will fail the template correlation test. Although this method increases the detection time and it is not guaranteed to work always, but it can be useful in some images where the background has a color similar to human skin. This method is similar to the method that was used by Rowley [13], however Rowley did it for the whole image and used a neural network to check that the component belongs to a face or not.

Images included in Figures illustrate our method on several images from the UCD database [18]. The first image is the original image. The second image is produced after applying the skin detection algorithm. The third image is after filling the holes and applying the morphological operations. The forth image shows the black and white image, as the background changes to black and the skin pixels to white. The fifth image shows each component with a color. The sixth image show placing the template on the component which has been considered as a face. The last (seventh) image is the final output of the proposed and implemented detection process.

In some images there may be two or three or more faces which are so close to each other that can become one component.

The method that we have introduced to detect faces in these cases is as follows:
1. Compute the height to the width of the component.

2. If the ratio is smaller than a threshold, then it means that the components may belong to more than one face. Due to the value of the ratio this component can consist of three or more faces; however it is not probable that a component consists of more than three faces.
3. Find the border of the two faces which are stuck together. For finding the border we count the number of pixels on the horizontal axes. The two maximums (if the threshold suggests there are two faces) are placed on the first and last third of the component, and the minimum on the second third of the component. The minimum is found and the two new components are now tested to find faces on them. If the correlations of these two new components (or more) are more than the correlation of the single component before splitting, then it means there were two (or more) faces in this component. This part is done so we can differ between two faces which are stuck together and a face which is rotated 90 degree. In this case the correlation of the whole component would be more than the correlation of the two new components, so the component will not be separated.

Face obeys the golden ratio, so the height to the width of the face is around 1.618. Sometimes with considering the neck which is mostly visible in images this ratio will increase to 2.4. So a height to width ratio between 0.809 and 1.2 shows that the component may belong to two faces. A value less than 0.809 shows that the component may consist of more than two faces. These values are calculated along the image orientation.

Figure 6 illustrate this situation. This image is not part of the UCD database. The same approach can be applied when the ratio of the height to the width is higher than a certain threshold. In this case it means that there may be two or more faces which are above each other. The same algorithm can be applied with some modification. Figure 7 shows this case. This image is not part of the UCD database. For two faces this threshold will be between 3.2 and 4.

Finding the border as mentioned in part 3 is faster and more accurate than using erosion (using as another method to separate two or more faces) because when using erosion, the process may need to be done several times so the two faces become separated, also in erosion while separating the two faces, some other parts of the image is also being eroded.

5. Conclusion

In this paper we have presented a novel methodology to detect faces on color images, with certain modifications in order to improve the performance of the algorithm. In some images when the faces are so close to each other that they cannot be separated after skin detection, we introduced a method for separating the face components. For the skin detection phase we used neural networks for recognizing human skin in color images. For future work, the face recognition phase can be added, where the faces which are detected and cropped can be recognized.

Unfortunately there is no database in this field. There are databases for face detection and databases for face recognition, but no database that covers both. So a database should be generated for this purpose.

AUTHORS

Mohammadreza Hajiarbabi – Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas, USA. E-mail: mehrdad.hajiarbabi@ku.edu

Arvin Agah * – Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas, USA. E-mail: agah@ku.edu

*Corresponding author

REFERENCES

[1] Al-Mohair H., Saleh J., Suandi S., "Human skin color detection: A review on neural network perspective", *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 12, 2012, pp. 8115–8131.

[2] Alshehri S., "Neural Networks Performance for Skin Detection", *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 12, 2012, pp. 1582–1585.

[3] Bakry E., Zhao H. M., Zhao Q., "Fast Neural implementation of PCA for face detection". In: *International Joint Conf. on Neural Networks*, 2006, pp. 806–811.

[4] Chandrappa D. N., Ravishankar M., RameshBabu D. R., "Face detection in color images using skin color model algorithm based on skin color information". In: *2011 3rd International Conference on Electronics Computer Technology(ICECT)*, 2011, pp. 254–258.

[5] Chen H., Huang C., Fu C., "Hybrid-boost learning for multi-pose face detection and facial expression recognition", Pattern Recognition, Elsevier, vol. 41, no. 3, 2008, pp. 1173–1185. DOI: http://dx.doi.org/10.1016/j.patcog.2007.08.010.

[6] Doukim C. A., Dargham J. A., Chekima A., Omatu S., "Combining neural networks for skin detection", *Signal & Image Processing: An International Journal* (SIPIJ), vol. 1, no. 2, 2010, pp. 1–11.

[7] Hajiarbabi M., Agah A., "Human Skin Color Detection using Neural Networks with Boosting", *Journal of Intelligent Systems*, under review, 2014.

[8] Hjelmas E., Kee Low B., "Face Detection: A Survey", *Computer Vision and Image Understanding* (CVIU), vol. 83, no. 3, 2001, pp. 236–274. DOI: http://dx.doi.org/10.1006/cviu.2001.0921.

[9] Jeng S., Liao H., Liu Y., Chern M., "An efficient approach for facial feature detection using Geometrical Face Model", *Pattern Recognition*, vol. 31, no. 3, 1998, pp. 273. DOI: http://dx.doi.org/10.1016/S0031-3203(97)00048-4.

[10] Kong W., Zhe S., "Multi-face detection based on down sampling and modified subtractive clustering for color images", *Journal of Zhejiang University*, vol. 8, no. 1, 2007, pp. 72–78.

[11] Kovac J., Peer P., Solina F., "Human Skin Color Clustering for Face Detection", EUROCON 2003. Computer as a Tool. The IEEE Region, vol. 8, no. 2, 2003, pp. 144–148. DOI: http://dx.doi.org/10.1109/EURCON.2003.1248169.

[12] Lin S. H., Kung S. Y., Lin L. J., "Face recognition/detection by probabilistic decision- based neural network", *IEEE Trans. on Neural Networks*, vol. 8, 1997, pp. 114–132.

[13] Rowley H., Baluja S., Kanade T., "Neural network-based face detection", *IEEE Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, 1998. pp. 22–38. DOI: http://dx.doi.org/10.1109/34.655647.

[14] Seow M., Valaparla D., Asari V., "Neural network based skin color model for face detection". In: *Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop* (AIPR'03), 2003, pp. 141–145.

[15] Shavers C., Li R., Lebby G., "An SVM-based approach to face detection". In: *2006 Proc. of 38th Southeastern Symposium on System Theory*, 2006, pp. 362–366. DOI: http://dx.doi.org/10.1109/SSST.2006.1619082.

[16] Singh S., Chauhan D. S., Vatsa M., Singh R., "A Robust Skin Color Based Face Detection Algorithm", *Tamkang Journal of Science and Engineering*, vol. 6, no. 4, 2003, pp. 227–234.

[17] Sung K., *Learning and example selection for object and pattern detection*, PhD Thesis, MIT AI Lab, 1996

[18] UCD database: http://ee.ucd.ie/~prag/

[19] Viola P., Jones M. J., "Robust real-time object detection". In: *Proc. of IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

[20] Wu Y., Ai X., "Face detection in color images using Adaboost algorithm based on skin color information". In: *2008 Workshop on Knowledge Discovery and Data Mining*, 2008, pp. 339–342.

[21] Yang G., Li H., Zhang L., Cao Y., "Research on a skin color detection algorithm based on self-adaptive skin color model". In: International *Conference on Communications and Intelligence Information Security*, 2010, pp. 266–270.

[22] Yang K., Zhu H., Y. J. Pan, "Human Face detection based on SOFM neural network". In: *2006 IEEE Proc. Of International Conf. on Information Acquisition*, 2006, pp. 1253–1257.

[23] Yang M. H., Kriegman D. J., Ahuja N., "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, 2002, pp. 34–58.