A Faster Face Detection Method combining Bayesian and Haar Cascade Classifiers

Érica K. Shimomoto, Almir Kimura, Jr., Ruan Belém

***Abstract*— Face Recognition is one of the most studied topics in computer vision. But before recognizing, it is necessary to detect a face. Many methods have been developed in order to perform this but the biggest problem they face is the time it takes to detect the object. This was greatly solved with the creation of the Haar Cascade Classifiers, which made possible the real time detection. However, depending on the application, it can still be time consuming. This work presents a faster way to detect faces by combining a Bayesian classifier with Haar Cascade classifiers.**

***Index Terms*—— Bayesian classifier, Face Detection, Haar Cascade Classifier, Machine Learning.**

I. INTRODUCTION

AUTOMATIC face recognition in images is one of the most studied topics in computer vision [1], since it is useful in people identification, surveillance tasks, intelligent environment, among others. Due to the extent research in the area, it is already possible to achieve high percentage of success in face recognition, under controlled environment [2].

But before recognizing a face, it is necessary to detect it. The face detection differs from the face recognition since its goal is no to give an identity to it, but simply discovering its position in an image.

The development of face detection techniques started around 1990 [3]. Among them, we can cite:

- Detection by color;
- Detection by motion;
- Detection in non-controlled environments.

Detection by color considers the RGB values of the colors that characterize the skin in order to find areas where there is a high probability of finding a face. By using filters and thresholds, it is possible to create a binary image where the possible skin areas receive the value 1, whereas the possible non-skin areas receive 0. This way, it is possible to analyze which possible skin areas have two holes, where, in general, indicate the eyes [4]. This is a simple and simple method, but it lacks robustness, once there is a great variety of a skin color.

Detection by motion considers the fact that a face is almost always moving (rarely is static in front of a camera). This was, it is possible to analyze which parts of an image move from one frame to another, indicating where the face is [5]. However, this technique will fail in case there are other objects, different from faces, moving in front of the camera. In order to reduce this error, it is possible to use the fact that human beings usually blink with both eyes at the same time. If you can find two motion areas with an acceptable distance between them (the distance between two eyes in a face), it is possible to have a higher detection rate [6][7].

As for detection in non-controlled environments, many artificial intelligence techniques have been used, such as Support Vector Machines and Neural Networks [8], but they can be time consuming, making them inapplicable for real time applications.

Apart from that, we can mention the Haar Cascade Classifiers, which use simple aspects in order to characterize a face [9]. This is a method that represents a revolution in the object detection history (not only faces) because it was the first fast enough method to be used in real time applications. These classifiers are trained with the AdaBoost[10] technique, using the Haar-wavelets features.

Since the Haar Cascade Classifiers can be used to detect many different objects, it is possible to, for example, train them to detect eyes, noses and mouths.

In many face recognition applications, it is necessary to have, not only the face position, but also the eyes, nose and mouth positions. All of this can be done using only the Haar Cascade Classifiers, but it can become really time consuming, especially if more than one face has to be detected and recognized at the same time.

Therefore, this work presents a faster way to perform that by combining two different face detection techniques to be described in this paper.

II. PROBLEM FORMULATION

The method to be described in this paper has the following goals:

1. Find the position of the faces in an image;
2. Find the position of the eyes, nose and mouth of each face;

We assume that:

1. The environment lighting is controlled and constant;
2. The place where the system will work is static;
3. The system will detect only frontal faces.

This work was submitted on September 20th, 2015.

E. K. Shimomoto is with the Universidade do Estado do Amazonas, Manaus, AM – Brazil (e-mail: erica_kido@yahoo.com.br).

A. Kimura, Jr., is with the Universidade do Estado do Amazonas, Manaus, AM – Brazil (e-mail: akimurajr@gmail.com).

R. Belém is with the TPV R&D Center Brazil, Manaus, AM – Brazil (e-mail: ruanjsb@gmail.com).

978-1-4673-8756-9/15/$31.00 ©2015 IEEE

III. Face Detector

One of the most applied methods for face detection is called Haar Cascade Classifier. In our method, the Detector must return a face object which consists of a face rectangle and the face elements center points (eyes, nose and mouth). Haar Cascade Classifiers can be used to detect those elements and, in order to do that, it is necessary to first find the face rectangle and then, inside this rectangle, search for the elements.

The Haar Cascade Classifier can be trained to recognize any object and therefore, the following method was proposed:

First, we need to find an area where there is a high probability of being a face. Considering that a face is mostly covered by skin and that skin has a very particular color, the technique of skin pixels segmentation was chosen for this task. Second, we validate these skin areas by searching for face elements using the Haar Cascade Classifier. If the elements exist and obey geometric restrictions, we have a face. Else, discard the skin area.

A. *Skin Segmentation*

In order to segment skin pixels, a Bayesian Classifier with histogram technique was selected. An image database with three classes was created:

1. Images with skin pixels only (12329 images, 20x20);
2. Images with non-skin pixels only (115 images, 1080x1920);
3. Images with skin pixels and non-skin pixels (153 images, 1080x1920);

The skin pixel images have small resolution because they were manually cut, in order to contain only skin pixels. The images with non-skin pixels only are images from sceneries, with no people in it. The images with skin pixels and non-skin pixels are pictures of people in different places.

The color space chosen was YCrCb because it allows us to separate the luma (luminance) component from the true color components, therefore being more robust to lighting changes [11]. Having all the pictures from the database represented in the YCrCb color space, we created a CrCb histogram for each image group. We will call them color model histograms.

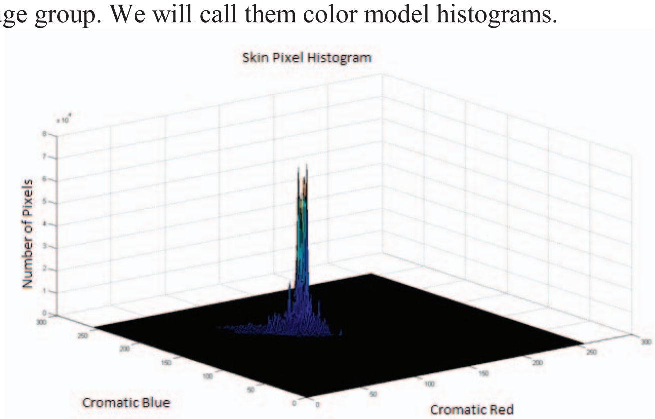

Fig. 1. Skin Pixel Histogram

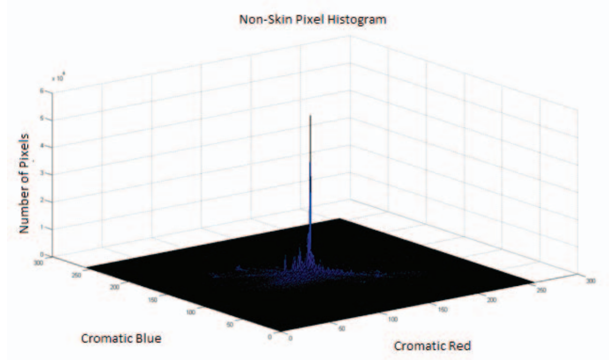

Fig. 2. Non-Skin Pixel Histogram.

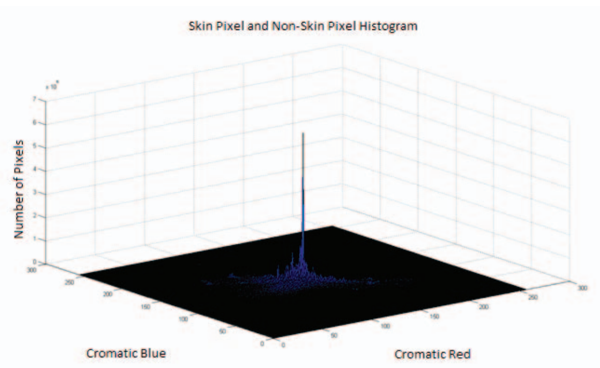

Fig. 3. Skin and Non-Skin Pixel Histogram.

Having these histograms, it is possible to make a small statistic study of the percentage of colors that exist in skin pixels against the set of all existing colors. Using the histogram technique with the Bayesian classifier, we can make the pixel classification in an image [12].

When we know the conditional probability of a pixel color in the YCrCb space, considering that it is a pixel with skin color:

$$P(CrCb*|skin) = \frac{[CrCb*]_{skin}}{Total_{skin}} \tag{1}$$

And the conditional probability of a pixel color in the YCrCb space, considering it is a pixel with non-skin color:

$$P(CrCb*|nSkin) = \frac{[CrCb*]_{nSkin}}{Total_{nSkin}} \tag{2}$$

We can use the Bayesian classifier, with the minimum cost decision rule [10]:

$$\frac{P(CrCb*|nSkin)}{P(CrCb*|skin)} \leq \alpha \tag{3}$$

Where:

- $[CrCb*]_{skin}$ is the number of pixels with the CrCb* color in the skin histogram;
- $[CrCb*]_{nSkin}$ is the number of pixels with the CrCb* color in the non-skin histogram;
- $Total_{skin}$ is the total number of pixels counted in the skin histogram;

- $Total_{nSkin}$ is the total number of pixels counted in the non-skin histogram;
- α is a threshold value.

If this ratio is smaller than α, it is pixel a skin pixel. Otherwise, it is a non-skin pixel.

There is no rule to define the value of α. It is directly related to the images used in each class. We empirically determine α, testing the skin segmentation in many skin areas and verifying the percentage of positive, false positive, negative and false negative results.

It is important to stress that the efficiency of this method depends on the quality of the database samples: Each class must have only pixels that belongs to that class.

This segmentation method resulted in a good precision in skin pixel detection and is really fast, since the probabilities are computed from values already stored in memory, acting as a mask.

Fig. 4. Skin Segmentation

B. *Face Elements Detection*

Once we have the skin areas, we need to verify the presence of the face elements: Eyes, nose and mouth. OpenCV computational vision library has a class especially created to deal with Haar Cascade Classifiers, called CascadeClassifier. OpenCV already has classifiers trained to identify eyes, nose and mouth.

In order to identify a skin segment found in the previous step, we verify the presence of two eyes, nose and mouth. We also verify a few geometric restrictions, such as:

- If the right eye x coordinate is smaller than the left eye x coordinate;
- If both eyes y coordinates are smaller than the nose y coordinate;
- If the nose y coordinate is smaller than the mouth y coordinate.

These restrictions assume that the face to be detected will never be upside down or with a high inclination.

If these restrictions are respected, we consider the skin area as a face.

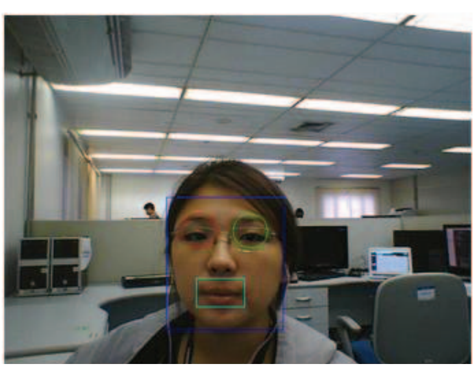

Fig. 5. Face Detected

If we look at the skin segmentation image, there are many small areas with skin pixels that cannot represent a face. Based on this, we can increase the efficiency of the process by discarding these small areas and verifying only the areas bigger than a set size.

The whole method can be described by the following flowchart:

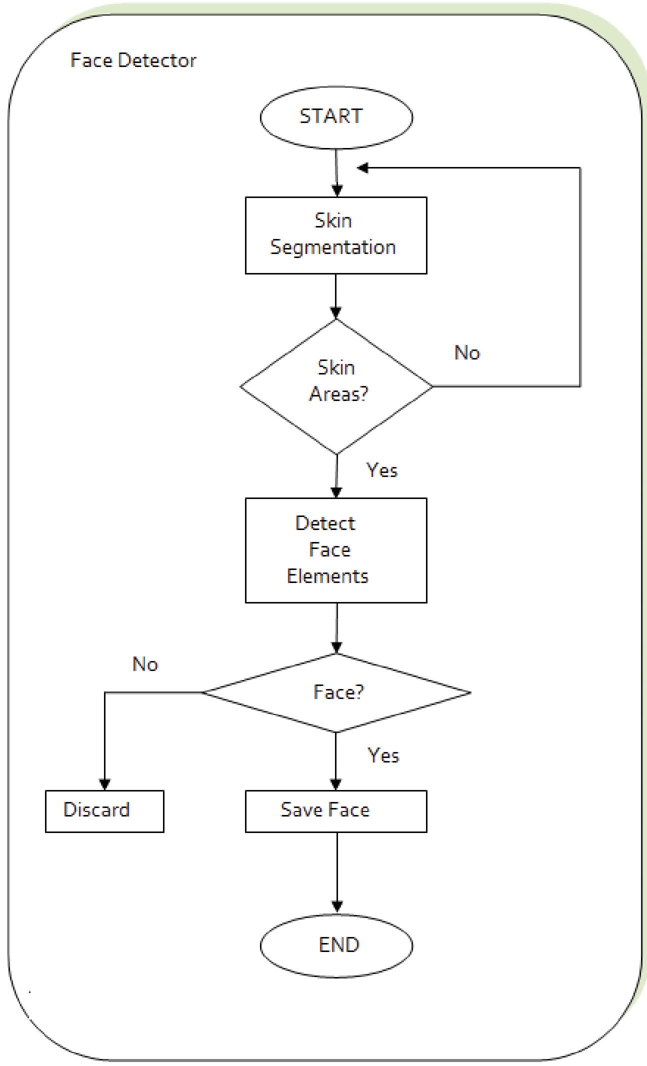

Fig. 6. Method Flowchart

IV. Results

A. *Skin Segmentation*

In order to implement the face detector, it was necessary to execute tests so we could define the parameter α for the skin segmentation process. By varying the value of this parameter, the segmentation becomes more biased to classify a pixel color as skin color or as non skin color.

We created a test database, following the same guidelines used for creating the training database. We then performed the skin segmentation with different values of α. By calculating the precision, sensitivity, specificity and accuracy, we created the following ROC (Receiver Operating Curve):

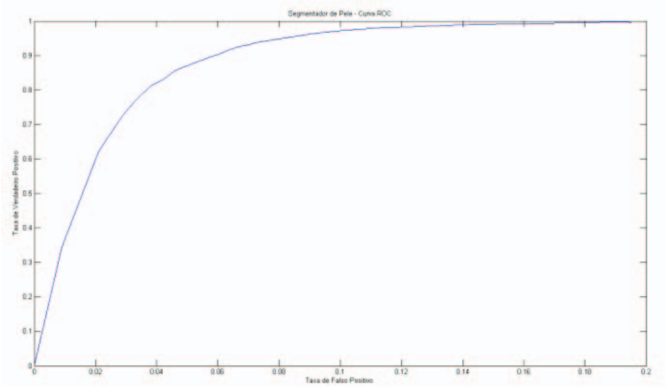

Fig. 7. ROC graphic for Skin Segmentation.

The value of α went from 0 to 30. The (0,0) point in the graphic was obtained with α. In this point, no matter the probabilistic distribution the color histograms returns, it will be always bigger or equal to zero and therefore all pixels will be classified as non skin color. As we increase the value of α, we start to have a better sensitivity to the skin color pixel and the best point for this model, which means, the closest to the point (1,0) in the graphic occurs when $\alpha = 0.4$.

After we defined the value of α, we performed the skin segmentation in 500 images from the LFW [13] database, using as selection criteria obtaining the following results:

TABLE I.
SKIN SEGMENTATION RESULTS

Skin Segmentation – α = 0.4	
Parameter	*Value (%)*
Precision	98.44
Sensitivity	82.91
Specificity	95.77
Accuracy	95.61

By analyzing the values, we observe that due the high specificity percentage, the skin segmentation performs a good identification for non-skin colors. On the other hand, the high sensitivity percentage shows that good skin color identification is performed.

The reason why false positive happens is because many colors that are considered skin colors can also be found in other surfaces. We can observe this confusion in the following image:

Fig. 8. Skin Segmentation – False Positive occurrence.

False negative also occurs, where the classifier fails to detect skin colors. Even though we used the YCrCb color space, it is not completely invariant to lighting variations and in some situations, the skin is not found:

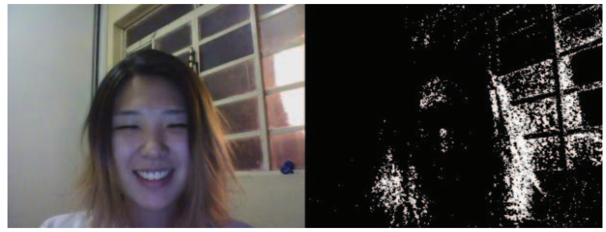

Fig. 9. Skin Segmentation – False Negative occurrece.

But overall, even with these mistakes, based on the Sensitivity and Specificity measurements, the skin segmentation has a good performance.

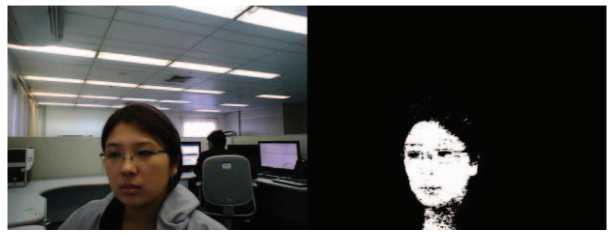

Fig. 10. Skin Segmentation – Successful segmentation.

B. *Face Detector*

In order to verify the face detector performance proposed by this work, two tests were made: One to verify the face detection itself and another to verify the speed of this method.

For the first test, we used 200 faces randomly chosen from the LFW database, different from the ones used in the skin segmentation test, plus 95 images with no face, took from the Internet, verifying its precision, sensitivity, specificity and accuracy. For comparison purposes, the same test was made using only Haar Cascade classifiers.

We considered faces correctly found as true positives, with the face elements also correctly found. Faces with elements wrongly found or images with no face, where it found a face, were considered as false positives. Samples with a face which returned no face by the detector were considered as false negatives.

The results are as following:

TABLE II.
FACE DETECTOR – USING ONLY HAAR CASCADE

Parameter	***Value (%)***
Precision	93.66
Sensitivity	69.63
Specificity	91.34
Accuracy	77.28

TABLE III.
FACE DETECTOR – USING PROPOSED METHOD

Parameter	***Value (%)***
Precision	90.18
Sensitivity	75.55
Specificity	88.26
Accuracy	78.30

Comparing both detectors, we can observe that the proposed method has some advantages over using only the Haar Cascade Classifiers. The method presented in this work has a higher sensitivity, which means that, from the positive samples in the database, it found more faces correctly than the Haar Cascade method. This reflects directly its accuracy.

However, using only Haar Cascade Classifiers resulted in a higher precision and specificity. Those differences exist, however they are small, showing that we managed to develop a face detection method with precision and accuracy comparable to the conventional method.

The second test was performed along with the first one, by measuring the detection time in its totality and in the elements detection. We only considered the time for correct detections. The average times are as following:

TABLE IV.
DETECTION TIMES USING HAAR CASCADE ONLY

Step	***Average (ms)***
Face Detection	310.532
Right Eye Detection	12.029
Left Eye Detection	8.740
Nose Detection	15.108
Mouth Detection	12.583
Total	358.992

TABLE V.
DETECTION TIMES USING PROPOSED METHOD

Step	Average (ms)
Skin Segmentation	3.449
Right Eye Detection	13.187
Left Eye Detection	8.905
Nose Detection	18.459
Mouth Detection	10.931
Total	54.931

The tests were run on a personal computer running Linux Ubuntu 12.04 Operational System, with 4GB RAM and an Intel i5 processor. The script was written in C++.

There is a big difference between the detection time averages of both methods. The proposed method detects the face, along with the elements, six times faster than only using the Haar Cascade Classifiers.

As for the performance parameters, we noticed that both methods have problems in detecting the face elements, since both use Haar Cascade Classifiers to execute this task. The Haar Cascade Classifier for eyes, available in the open library OpenCV, showed a good performance in finding frontal or almost frontal faces.

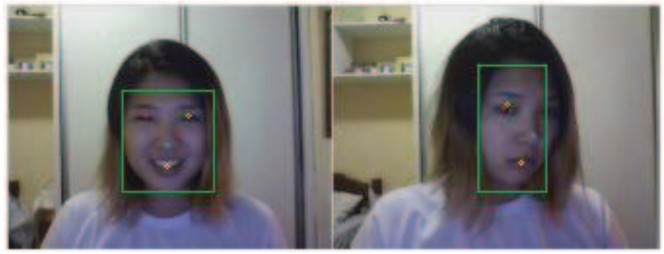

Fig. 11. Detection by the proposed method. a) True positive b) False positive.

However, for profile faces, it is usual for the detector to find both eyes in the same side of the face. The b) image from figure is a false positive because, although the detector returned a face object, it failed in detecting the eyes.

It is also necessary to emphasize that both detectors fail when the person wears sun glasses. The detectors also show some difficulties in finding faces with thicker frames.

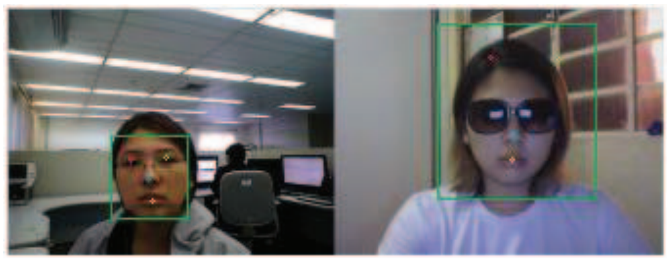

Fig. 12. Detection with the proposed method – Detecting faces with glasses

Based on these two tests, we can see that it was possible to develop a new face detection method applied to this work problem, faster than the conventional method without interfering in the detection rate.

V. FINAL CONSIDERATIONS

In this work, a faster face detection method was proposed, based in the skin color segmentation, aiming the reduction of execution time.

As for the skin segmentation, although there is some confusion when non-skin pixels present a color similar to a skin color, the number of false negatives is small when compared to the number of true positives. Using the YCrCb space color was efficient in obtaining the color histograms, leading to good skin color pixel detection in various environments found in the test database used.

As for the detector, the biggest problem is in the face elements detection. It is necessary to rather change the geometric restrictions so we don’t get a face with both eyes on the same side of the face, or look for a different detection method. Considering that in this application the person will be facing directly the camera, finding both eyes on the same side of the face is unlikely to happen. Once this system will be in a fixed place, it is possible to manipulate the environment so there are no objects with color similar to skin colors.

And although all these problems happened, it was possible a face detector that is, in average, six times faster than the detector that uses only the Haar Cascade Classifiers.

VI. Future Work

There is a great number of possibilities for future work. Although the results were promising, the system still is not good enough for real applications. Among the improvements to be performed, we can mention:

- Reduce the number of false positive in the skin segmentation: Increase or modify the database used to obtain the color histograms. Improve the generalization of colors by approximating the curve in the histograms to a Gaussian. The only point that is necessary to be careful with is the increase of computational processing when calculating the Gaussian.
- Improve the detection of face elements: Use or create a new method for elements detection;
- Perform more exhaustive testing, in order to better evaluate the detector performance.

Acknowledgment

We would like to thank Professor Walter Valenzuela from the University Of Amazonas State (UEA), whose opinion and critics motivated the efforts in making this project a success. Also to the TPV R&D Center Brazil team who gave valuable opinion and feedback that helped in the development of this work. And last, but not least, to Denso Industrial da Amazônia Ltda. for the great support.

References

[1] C. Zhang and Z. Zhang, "A survey of recent advances in face detection", Tech. rep., Microsoft Research, Tech. Rep., 2010.

[2] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on, vol. 1. IEEE, 2005, pp. 947–954.

[3] G. Mislav and K. Delac. (2007) Face Recognition. [Online]. Available: http://www.face-rec.org/general-info/

[4] W. F. H. CHANG and U. ROBLES. (2000) Face detection. [Online]. Available: http://www-cs-students.stanford.edu/ robles/ee368/main.html

[5] P. REIGNIER. (1995) Finding a face by blink detection. [Online]. Available: http://www-prima.imag.fr/ECVNet/IRS95/node13.html

[6] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007, pp. 1–8.

[7] R. W. FRISCHHOLZ. (2014) The face detection: Techniques. [Online]. Available: http://www.facedetection.com/facedetection/techniques.htm

[8] C. Shan, "Face recognition and retrieval in video," in Video Search and Mining. Springer, 2010, pp. 235–260.

[9] P. Viola and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.

[10] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," Machine learning, vol. 37, no. 3, pp. 297–336, 1999.

[11] R. W. G. Hunt, M. R. Pointer, and M. Pointer, Measuring colour. John Wiley & Sons, 2011.

[12] S. L. Phung, A. Bouzerdoum, and D. Chai Sr, "Skin segmentation using color pixel classification: analysis and comparison," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, no. 1, pp. 148–154, 2005.

[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.