Skin detection in images using machine learning techniques for color and texture recognition

Stefan Sebastian

May 4, 2018

Abstract

The interfaces between human and computer interaction have evolved from simple buttons to voice and video recognition. Skin detection is a useful preprocessing step for detecting people in images, which facilitates this communication. This article presents a method of detecting skin in images by combining several state of the art approaches in this domain. The model combines Skin Probability Maps for color analysis on images segmented with the Quick shift algorithm and texture detection using Haralick's features. The results, evaluated on the widely used Compaq dataset, are comparable to others found in literature: 80% true positive rate and 20% false positive rate.

Contents

1 Introduction

Skin detection means identifying pixels and regions in images which correspond to human skin. Starting from these regions we can build models that detect people or certain body parts.

The purpose of this paper is to present the best models currently employed in skin detection and to propose a new model that combines some state of the art detectors. To the best of my knowledge no method that combines image segmentation, color and texture analysis has been published in this area of research.

2 Applications

One of the first applications in this field of study was scanning internet images in order to filter pornographic content [1]. The model proposed in that paper scans the images for large areas of skin and then applies some geometrical analysis looking for elongated shapes.

Another application was detecting anchorpersons in talk shows [2] for automatic annotation and storage. The data can be used for quickly finding videos containing specific persons or calculating how much time each anchorman was on stage.

Skin detection is usually an intermediary step for detecting more concrete body parts, so I will also present some applications of those methods. Face and hands detection is used to facilitate human computer interaction. By scanning gestures and reactions a software system can respond in a more convenient way to the user. For example the Kinect camera, which connects to the Xbox console, allows the user to navigate menus using hand swipes [3]. Face recognition, the next step after detection, is useful for automatic identification of people [4] which has applications in security, such as unlocking your phone only when it scans your face, and monitoring certain areas, like the systems used by the Chinese government to fine jaywalkers caught on camera without human intervention [5].

3 Related work

The subject of skin detection has been of interest for researchers due to its many applications.

3.1 Skin Probability Maps

One of the most comprehensive experiments in skin detection was conducted by MJ Jones and JM Rehg [6] who created the Compaq skin dataset, which became a standard for evaluating results of research in this domain.

They used skin probability maps, which are histogram based models, meaning they set a number of bins for each color channel where they keep track of how many times each color pixel appears in the training set. Although 256^3 would be an obvious choice for the number of bins (one for each pixel) they determined that 32^3 is the optimum due to a large number of pixels not appearing in the training set. Each pixel is classified as being skin or non-skin using Bayes' theorem $P(rgb|skin) >= \beta$, where β is the threshold value. For this model they used a 6822 photo training set, where 4483 photos contained no skin and 2336 had portions of skin. The testing set had similar dimensions: 6818 total photos (4482 non-skin and 2336 skin). Using this method they obtained a 80% true positive rate and 8.5% false positive rate or, with a different threshold, a 90% TP rate and 14.2% FP rate.

3.2 Gaussian Models

Gaussian Models are probabilistic methods of representing skin distribution using Gaussian probability density functions. They rely on the assumption that skin pixels cluster in a small area of a color space [7].

The previous paper [6] also proposes a combination of two mixture models for skin and non-skin classes. Each model is composed of 16 gaussians and was trained on approximatively 74% of the histogram data, because only that data was available at the time. The results were similar to those of the previous model: 80%/9.5% and 90%/15.5%.

Lee and Yoo [8] presented a Single Gaussian in CbCr space and a Gaussian Mixture in IQ space, both trained and tested on the Compaq dataset with the results of 90%/33.3% and 90%/30%, respectively.

3.3 Explicit thresholding

Thresholding is one of the fastest and simplest methods for skin detection. The basic idea is to define a set of rules and thresholds for the values of pixels in a given color space. This approach is best suited for real-time detection

due to its speed. Some examples include [9], [10], [11], [12] for face detection systems in different color spaces with varying results.

A set of thresholds for YCrCb space where proposed by Chai and Ngan[10]. They set the ranges for Cb from 77 to 127 and for Cr from 133 to 173 and worked with the ECU database.

Dai and Nakano[11] created a model for YIQ, an orthogonal color space, which only used the I component (which stands for in-phase). The range they provided was [0, 50], however most of the images in their databases where of people with yellow skin.

Brand and Mason [13] applied the technique from [12], which uses the YI'Q' space with the following threshold $14 < I' < 40$, on the Compaq dataset and obtained 94.7% TP rate and 30.2% FP rate.

An interesting approach, proposed by Gomez and Morales[14], is having a learner find these rules automatically. They use RCA, a constructive induction algorithm, to build rules expressed with simple arithmetic operations in the rgb space. Their method achieves better results than the Bayesian SPM on their dataset, however it is computationally slower. The strategy implemented for RCA was finding attributes which cover either a large number of true positives or a few false positives. The starting attributes where r, g, b and the constant 1/3, which would generate new attributes using the operators : +, *, - and squaring. One of the best performing and simplest looking of the generated models looks like this:

$$\begin{aligned} \frac{r}{g} &> 1.185 \quad \text{and} \\ \frac{r*b}{(r+g+b)^2} &> 0.107 \quad \text{and} \\ \frac{r*g}{(r+g+b)^2} &> 0.112 \end{aligned} \tag{1}$$

In comparison with the C4.5 decision tree algorithm, the RCA method obtained slightly worse results but with much simpler rules. They used a custom dataset containing images of more than 2000 people and obtained around 90% both in precision and recall.

3.4 Models using segmentation

Frerk and Al-Hamadi[15] proposed a method that applies image segmentation combined with a Bayesian SPM. The first step is calculating the probability

for each pixel in an image and creating P_I, the pixel probability image. P_I is used as input for a SLIC algorithm that calculates the superpixels. A probability is then computed for each superpixel and compared to a threshold.

In [16] a similar approach to [15] is taken. Firstly, a segmentation is performed and superpixels are extracted. A probability is calculated for each superpixel as the average of its component pixels probabilities. Then, a CRF(Conditional Random Field) method is used in order to obtain smoother skin regions. This model was tested on the Compaq database and obtained a 91.17% TP rate and 13.12% FP rate.

3.5 Models using textural features

[17] presents a model based on Artificial Neural Networks that analyzes both color and textural features. The color features used are the mean color, the standard deviation and the skewness. These are calculated for each channel (R, G, B). The inputs for the ANN also include the following textural information: Entropy, Energy, Contrast and Homogeneity, which are computed from the Gray-Level Co-Occurrence Matrix. The networks has three layers: an input layer, a hidden layer (with 50 neurons) and an output layer. The model was trained on 300 images (80x80 px) of skin and non-skin textures and tested on 100, obtaining a 96% accuracy in classifying whether an image is a skin patch or not.

Medjram et al.[18] proposed another method that combines color and texture information. The first step is converting the initial image to the YCbCr color space. Skin regions are identified using thresholding like the following: $77 <= Cb <= 127, 133 <= Cr <= 173$. Secondly, the image is sharpened in order to enhance its texture. The features considered from the GLCM are Contrast, Homogeneity and Energy, calculated over a 5x5 matrix. The last step is applying a Support Vector Machine classifier on the initial image to classify texture patches.

In [19] they use Gabor wavelet transforms to compute textural attributes. After identifying the initial skin regions a watershed segmentation algorithm is applied to increase true acceptance rate.

3.6 Hidden Markov Models

Sigal et al.[20] implemented a method for real time skin detection in videos. Their model predicts the evolution of the skin color histogram using a sec-

ond order Markov model, using an initial prediction from a Bayes classifier over the Compaq dataset. It uses the EM algorithm which consists of two steps: E(frame segmentation based on histogram) and M(histogram adaptation based on feedback from the current frame).

4 Proposed model

The model proposed by this paper is made up of three steps. The first step is image segmentation, which aims to divide the input image into several regions, called superpixels, based on color and shape. A Bayesian probability is then calculated for every pixel in the region and the average of these is compared to a threshold to determine if the superpixel's color is likely to be that of skin. Finally, a Support Vector Machine analyzes patches around each pixel and calculates Haralick features in order to classify the texture. The results from color and texture detection are then combined into the final result.

4.1 Datasets

For this experiment I have used two databases. First, the Compaq dataset [6], which is one of the most cited in literature, contains 13.000 images out of which 4700 contain skin and was used for the skin color model. The images were downloaded using a web crawler then divided into skin and non-skin images. For every skin image a black and white mask was created by hand to mark the regions of interest.

The second dataset, SFA [21], was created as a combination of FERET(876 images) and AR(242 images) datasets. It contains skin and non-skin patches of dimension from 1x1 to 35x35, which makes it ideal for the texture model. For each dimension, SFA has 3354 skin images and 5590 non-skin images.

4.2 Segmentation

Image segmentation has been applied to increase smoothness and eliminate holes in regions classified as skin. The algorithm chosen for image segmentation is Quick shift due to its simplicity, speed and its ability to form clusters without knowing their number beforehand.

A suitable feature space for image segmentation is the combination of RGB components and pixel position, which can be scaled[22]. The algorithm starts by calculating a Parzen density value for each data point. Then it links each pixel to the nearest neighbor with a higher density. To obtain the regions we can limit our search during linking phase to a distance of τ. When calculating the density we can limit our search to a 3σ window because the contributions for pixels further away should be small [22].

```
# density computation;
for x in all pixels do
    P[x] = 0;
    for n in all pixels less than 3 * σ away do
        P[x] += exp(-(f[x] - f[n])^2 / (2 * σ * σ))
    end
end
# neighbor linking;
for x in all pixels do
    for n in all pixels less than τ away do
        if P[n] > P[x] and distance(x, n) is smaller than to previous
         parent then
            d[x] = distance(x, n);
            parent[x] = n;
        end
    end
end
```

Algorithm 1: The Quick shift segmentation algorithm from [22]

4.3 Color analysis

Each superpixel is evaluated with a color detection model. The chosen model is a Bayesian Skin Probability Map, due to its simplicity and speed. After scanning the input images we can calculate the following features: the number of skin pixels and non-skin pixels and the number of apparitions as skin and non-skin for each pixel. To determine whether a pixel can be classified as skin we can apply Bayes' Theorem as follows: $p = P(X|S) * P(S)/P(X)$, where P(S), P(X) and P(X|S) represent the probabilities of finding a skin pixel, the selected pixel and the selected pixel given skin, respectively.

The authors of the Compaq [6] dataset made the observation that 77% of the RGB space is empty and as a solution computed the probabilities on groups of pixels. The proposed model uses a similar method by taking the maximum probability in an area around the given pixel. This ensures that the considered pixel is always at the center of the group.

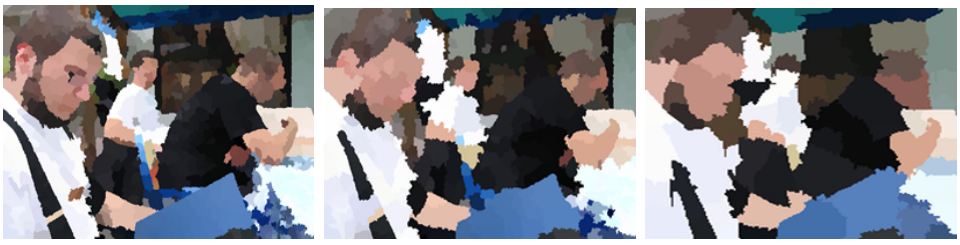
4.4 Texture analysis

Texture analysis is an independent step whose result is combined with that of the color detector's. Texture, while easily identifiable by a human observer, does not have precise definition. Consequently there are many choices for building a texture classification model depending on the author's interpretation.

The chosen method for this model is the one proposed by Haralick [23]. Features that describe texture are extracted from a matrix called the Gray Level Co-occurence Matrix, which represents how often combinations of pixel gray levels at different offsets appear in an image. A Support Vector Machine is trained on these features extracted from 5x5 patches. Each pixel is classified by building a window of the same size around it and giving that patch as input to the classifier.

4.5 Parameters

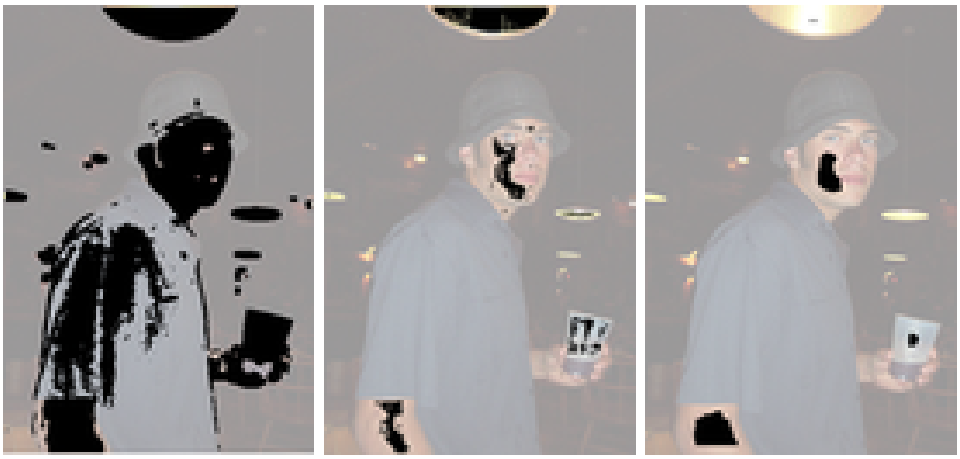
For image segmentation I tried to find values for the sigma and tau parameters that produce regions as large as possible without distorting the contours. I chose the (3, 5) pair, which can be seen in comparison with other pairs in figure 1.



(a) $\sigma = 3, \tau = 5$ (b) $\sigma = 2, \tau = 10$ (c) $\sigma = 4, \tau = 8$

Figure 1: Segmenting an image with different parameters

The most suitable area for texture analysis was also determined experimentally. As can be observed in figure 2 the 5x5 window offers a good detection of skin areas while the true positive rate goes down drastically for larger ones.



(a) 5x5 (b) 7x7 (c) 35x35

Figure 2: Texture detection with different windows

I tried building color models in RGB, HSV and YCrCb and concluded that Bayesian color analysis obtains roughly the same results regardless of the color space chosen. For every pixel, I chose the maximum probability on a distance of 4 (similar to the bin size used in [6]) on every feature. The threshold value can be used to control the TPR/FPR ratio.

4.6 Evaluation

The model was evaluated on the Compaq skin database in order to compare it with other approaches from literature. There is a difference however, in the number of images chosen for evaluation. While most references use a 6000 image set for testing I used only 50 images due to time constraints.

For each image in the testing set I calculated the True Positive Rate and the False Positive Rate by comparing each pixel from my output to the given mask and then I averaged the results.

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{2}$$

Equation 2 shows how those metrics were calculated. TP, FP, TN and FN stand for the numbers of true positives, false positives, true negatives and false negatives, respectively.

Table 1: Evaluation of the proposed model on the Compaq dataset

TPR	**FPR**	**Threshold**
86.57%	28.48%	0.1
76.29%	15.54%	0.25
81.59%	21.29%	0.167

5 Conclusions

The model obtains results similar to the ones presented in literature however image segmentation and analyzing probabilities for every pixel's neighbors take a toll on computation speed. Due to time limitations, the evaluation was done on a small subset of the data. Consequently the model's performance might improve over a larger set of input data.

References

[1] M. M. Fleck, D. A. Forsyth, and C. Bregler, "Finding naked people," in *Computer Vision — ECCV '96* (B. Buxton and R. Cipolla, eds.), (Berlin, Heidelberg), pp. 593–602, Springer Berlin Heidelberg, 1996.

[2] B. M. and R. Amsaveni, "Anchor person detection using haar-like feature extraction from news videos," *International Journal of Computer Applications*, vol. 153, no. 9, pp. 23–27, 2016.

[3] "The complete list of kinect gesture and voice commands for your referencing pleasure." `https://news.xbox.com/en-us/2013/11/26/xbox-one-kinect-gesture-and-voice-guide/`. Accessed: 2018-05-03.

[4] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 24, no. 1, pp. 34–58, 2002.

[5] C. Geib, "If you jaywalk in china, facial recognition means youll walk away with a fine." `https://futurism.com/facial-recognition-china-social-credit/`. Accessed: 2018-05-03.

[6] M. J. Jones and J. M. Rehg, "Statistical color models with applications to skin detection," *International Journal of Computer Vision*, vol. 46, pp. 81–96, 2002.

[7] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition*, vol. 40, pp. 1106–1122, 2007.

[8] J. Y. Lee and S. I. Yoo, "An elliptical boundary model for skin color detection," in *In Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology*, 2002.

[9] J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection," *EUROCON*, 2003.

[10] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE TRANSACTIONS ON CIRCUITS*

AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. 9, no. 4, pp. 551–564, 1999.

[11] Y. Dai and Y. Nakano, "Face-texture model based on sgld and its application in face detection in a color scene," *Pattern Recognition*, vol. 29, no. 6, pp. 1007–1017, 1996.

[12] C.Wang and M.Brandstein, "Multi-source face tracking with audio and visual data," *IEEE MMSP*, p. 168, 1999.

[13] Brand, J., and Mason, "A comparative assessment of three approaches to pixellevel human skin-detection," *In Proc. of the International Conference on Pattern Recognition*, vol. 1, pp. 1056–1059, 2000.

[14] G. Gomez and E. F. Morales, "Automatic feature construction and a simple rule induction algorithm for skin detection," *Proceedings of Workshop on Machine Learning in Computer Vision*, pp. 31–38, 2002.

[15] F. Saxen and A. Al-Hamadi, "Superpixels for skin segmentation," *Workshop Farbbildverarbeitung, At Wuppertal*, vol. 20, 2014.

[16] R. P. Poudel, J. J. Zhang, D. Liu, and H. Nait-Charif, "Skin color detection using region-based approach," *International Journal of Image Processing (IJIP)*, vol. 7, no. 4, 2013.

[17] N. K. E. Abbadi, N. Dahir, and Z. A. Alkareem, "Skin texture recognition using neural networks," *CoRR*, vol. abs/1311.6049, 2013.

[18] M. Sofiane, B. M. Chaouki, and M. B. Yamina, "Improved skin detection using colour space and texture," *International Journal of Computer and Information Engineering*, vol. 8, no. 12, 2014.

[19] Z. Jiang, M. Yao, and W. Jiang, "Skin detection using color, texture and space information," *Fuzzy Systems and Knowledge Discovery*, 2007.

[20] L. Sigal, S. Sclaroff, and V. Athitsos, "Skin color-based video segmentation under time-varying illumination," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 26, no. 7, 2004.

[21] J. P. B. Casati, D. R. Moraes, and E. L. L. Rodrigues, "Sfa: A human skin image database based on feret and ar facial images," *IX Workshop de Viso Computacional, Rio de Janeiro*, 2013.

[22] B. Fulkerson and S. Soatto, "Really quick shift: Image segmentation on a gpu," in *In Proceedings of the Workshop on Computer Vision using GPUs, held with the European Conference on Computer Vision*, 2010.

[23] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, vol. 3, no. 6, 1973.