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

Stefan Sebastian

April 29, 2018

Contents

Chapter 1

Theoretical background

1.1 Related work

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

1.1.1 SPM

One of the most comprehensive experiments in skin detection was conducted by MJ Jones and JM Rehg[1] 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 methods 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.

1.1.2 Gaussian Models

The previous paper[1] 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%.

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

1.1.3 Explicit thresholding

Another popular approach is explicit thresholding. Some examples include [3], [4], [5], [6] for face detection systems in different color spaces with varying results.

A set of thresholds for YCrCb space where proposed by Chai and Ngan[4]. 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[5] 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.

An interesting approach, proposed by Gomez and Morales[7], is having a learner find these rules using the RCA algorithm.

Brand and Mason [8] applied the technique from [6], 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.

1.1.4 Models using segmentation

Frerk and Al-Hamadi[9] 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 [10] a similar approach to [9] 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.

1.1.5 Models using textural features

[11] 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.[12] 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 [13] 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.

1.1.6 Hidden Markov Models

Sigal et al.[14] implemented a method for real time skin detection in videos. Their model predicts the evolution of the skin color histogram using a second 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).

1.2 Image segmentation

Image segmentation is a technique for dividing an image into several regions that contain similar pixels. These partitions are often called super-pixels and represent an abstraction layer over the initial image. They can be characterized by color, border or shape(circle, ellipse, polygon, etc.)[15]. The main purpose of segmentation is to represent areas of interest in an image such as faces, fields, buildings, etc, and usually serves as a preparation step for a more complex detection algorithm.

Ideally, the resulting regions should have the following characteristics: uniformity according to the selected feature, such as color or texture, a small number of holes(subregions that differ considerably from the container region), a notable difference from the neighboring areas and smooth borders.[15].

This problem has been researched extensively over the years however there is not a single best solution available. I will present some of the algorithms presented in literature.

1.2.1 Thresholding

Thresholding is one of the simplest approaches to image segmentation. It consists of finding a threshold T for the gray level of pixels in the image. Therefore, we can classify every pixel by comparing its brightness to T. This technique is a perfect fit for separating objects from a darker background but has severe limitations in other tasks[16].

One of the problems with global thresholding, choosing a single value for T over the whole image, is dealing with different levels of illumination. For that reason the local thresholding approach was developed. This expands the previous method by using multiple thresholds for different parts of the image. Chow and Kaneko [17] applied a similar method for detecting the left ventricle of the heart in x-ray pictures. They divide the original image in multiple blocks that do not overlap and calculate a threshold for each of them.

Threshold selection can be done manually or automatically and usually involves analysis of a gray level histogram where the size of each bar is proportional to the number of pixels with that brightness[18]. For example, in an image that contains some objects in front of a darker background the histogram is likely to contain two peaks separated by a valley and we can choose T somewhere in between those peaks.

Automatic threshold detection can be done by modeling object and background populations with a normal distribution like the experiments presented in [19]. The method proposed calculates a least-squares fit of a function f(i), where i is the gray level, to the histogram, using a hill climbing algorithm. The function takes into account the mean and standard deviation of the histogram. Lastly, the best fitting f(i) is the used for classification.

However simple, all these methods have limitations, such as not considering spatial factors, and having no control over border smoothness and holes inside detected regions.

1.2.2 Edge detection

These methods aim to solve image segmentation problems by detecting all edges in images. An edge is characterized by a sudden change in pixel intensity[16]. The result of edge detection is an image that represents to classes of pixels: part of an edge or not. I will present some of the most popular methods.

The Roberts Detection[20] is one of the fastest methods due to its simplicity. It uses the following convolution masks:

$$Gx = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} Gy = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}$$

Each of theses calculates a gradient value for an orientation and their values can be combined to determine an absolute magnitude of the gradient using: $|G| = \sqrt{G_x^2 + G_y^2}$. Lastly, we can apply thresholding for the resulting value to determine if it is part of an edge or not.

Other similar detectors are the Prewitt and the Sobel detectors[20]. The difference is that they use a larger convolutional mask, 3x3, and in case of the former, more orientations.

Some soft computing techniques such as: neural networks, genetic algorithms and fuzzy logic[20], have also been applied in edge detection problems. However they don't offer significant improvements in performance over the simpler and faster methods.

1.2.3 Clustering

We can view the image segmentation process as a clustering problem. All pixels are associated with a region, which corresponds to a cluster.

K-means

One popular approach to segmentation problems is the K-means algorithm, which can divide n data points in k clusters, where k is known in advance. This can be applied to image segmentation if we consider pixel intensities as the values on which we are doing clustering. The algorithm is simple and efficient on large volumes of data[16], however the number of clusters must be known in advance, which can be problematic as the number of distinct regions can vary from image to image.

Mode seeking. The Quick shift algorithm

Mode seeking algorithms are methods for clustering without needing to know the number of clusters in advance. The first step of these algorithms, as presented in [21], is calculating a Parzen density estimate over all given points: $P(x) = \frac{1}{N} \cdot \sum_{i=1}^{N} \cdot k(x - x_i)$. The next step is moving each point towards a mode of P(x) on a trajectory determined by the gradient of P. Clusters are formed by the points converging on the same modes.

The Quick shift algorithm[22] is one of the most efficient in this family. It starts by calculating a Parzen density estimate, most often using an isotropic Gaussian window:

$$P(x) = \frac{1}{2\pi\sigma^2 N} \cdot \sum_{i=1}^{N} e^{\frac{-\|x-x_i\|^2}{2\sigma^2}} \tag{1.1}$$

Every point is then linked to the nearest one with a higher density and the resulting structure is a tree. In order to get our regions we can either cut branches with a size larger than τ or analyzing data points less then τ distance away on the linking step. If we choose the latter the result will be a collection of trees, each representing a cluster.

In the application of Quick shift in image segmentation a relevant feature space must be chosen. An option is using the RGB components and position(x, y coordinates) of each pixel. As the position does not always has fixed bounds (image size can vary) we can scale the position components depending on our input data, such that the importance of color and position remains similar[22].

An example pseudocode implementation as presented in [22] looks like this: When calculating the density we 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
```

Some advantages of Quick shift are simplicity of implementation, speed ($O(N^2)$), ability to work on any type of data and the control of fragmentation with the given parameters[21].

1.3 Skin detection by color

Skin pixel detection by color means classifying a pixel while considering only its color features. A first step in applying this approach is selecting a color space.

1.3.1 Color spaces

A color space, also called a gamut, represents a set of colors in a way that is independent of the medium in which they are represented(computer screens, cameras, magazines, etc)[23]. The L*a*b* color space contains all colors that can be seen by the human eye, however most color spaces are smaller due to technical limitations. I will present some of the color spaces which have been used successfully to classify skin pixels.

RGB

To start with, RGB is one of the most popular color spaces for working with image data. It matches the color sensitive receptors of the human eye(red, green, blue) and started as a convenient way to represent the colored rays used by CRT screens[24]. While this model is simple to use it has the disadvantage of mixing chrominance and luminance features[24].

Normalized RGB

Normalized RGB is a color space with a lighter memory consumption than RGB and its components are calculated as follows[24]:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}. \tag{1.2}$$

The third value can be determined from the other 2 so we can avoid storing it. Other advantages according to [25] include reduced differences caused by illumination and ethnicity, and lower variance of skin color clusters than in the normal RGB space.

HSI, HSV, HSL

HSI, HSV, HSL represent perceptual color spaces and they describe the hue, saturation and intensity (or value, lightness). These color spaces are used

because they provide invariance to ambient lighting and surface orientation relative to the source of light[25]. We can convert to HSV from RGB using the following formulas[24]:

$$H = arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B))}} \tag{1.3}$$

$$S = 1 - 3\frac{min(R,G,B)}{R+G+B} \tag{1.4}$$

$$V = \frac{1}{3}(R+G+B) \tag{1.5}$$

YCbCr

Orthogonal color spaces, which YCbCr is a member of, provide chrominance and luminance separation as they represent colors with statistically independent components. YCbCr is mostly used by European television studios and in image compression[24]. Y represents luma (or luminance) and Cb, Cr are the blue and red difference chroma components and they can be computed as follows:

$$Y = 0.299R + 0.587G + 0.114Bs \tag{1.6}$$

$$Cb = B - Y \tag{1.7}$$

$$Cr = R - Y \tag{1.8}$$

Having such a simple transformation and a clear separation of the luminance component makes the YCbCr a popular choice for skin detection models.

1.3.2 Explicit thresholding

This is one of the simplest skin-color models that can be built. The method aims to define, through the use of simple rules and thresholds, the boundaries of skin clusters in a specific color space. It has been observed in [26] that the colors of human skin tend to cluster in small regions of the color space and human skin pixels differ more in intensity than in color.

An example using the RGB space, from Peer et al.[3] which has been integrated into a face detection system consists of the rules below:

$$\begin{aligned} R > 95 \quad \text{and} \quad G > 40 \quad \text{and} \quad B > 20 \quad \text{and} \\ max\{R,G,B\} - min\{R,G,B\} > 15 \quad \text{and} \\ |R - G| > 15 \quad \text{and} \\ R > G \quad \text{and} \quad R > B \end{aligned} \tag{1.9}$$

These thresholds and rules can also be generated using a machine learning algorithm, such as the one proposed by Gomez and Morales[7]. 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. RCA stands for Restricted Covering Algorithm which resembles a general covering algorithm with the restriction of trying to build a single rule for each class (in this case, a rule for skin detection). 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 their best and simplest generated models looks like this:

$$\begin{aligned} \frac{r}{g} > 1.185 \quad \text{and} \\ \frac{r * b}{(r+g+b)^2} > 0.107 \quad \text{and} \\ \frac{r * g}{(r+g+b)^2} > 0.112 \end{aligned} \tag{1.10}$$

In comparison with the C4.5 decision tree algorithm, the RCA method obtained slightly worse results but with much simpler rules.

1.3.3 Skin Probability Map

A SPM represents a histogram with multiple bins. Each bin stands for a color or a subset of colors and has a value equal to the probability of holding skin colored pixels. When building a SPM you must choose the color space and the number of bins per color channel.

In order to determine whether a given pixel is a skin colored pixel we apply Bayes' theorem. Here is the form used by Jones and Rehg[1] in one of the most popular papers on statistical skin detection:

$$P(skin|p) = \frac{P(p|skin)P(skin)}{P(p|skin)P(skin) + P(p|\neg skin)P(\neg skin)} \tag{1.11}$$

In this equation "p" is the notation for the occurrence of the given pixel. Therefore P(skin—p) means the probability of observing skin given our pixel. To determine whether we classify the pixel as skin we compare our probability with the threshold value, β.

$$P(skin|p) > \beta \tag{1.12}$$

The probability of observing skin can be computed as the ratio of the number of skin pixels to the total number of pixels observed in training.

$$P(skin) = \frac{T_S}{T_S + T_N} \tag{1.13}$$

Jones and Rehg[1] made the observation that given even a large training set most pixels are never seen. They explored their dataset of approximatively 2 billion pixels and came to the conclusion that around 77% of the RGB space is empty. This suggests that we might get better results by reducing the number of bins per channel. We can observe that a small perturbation in the RGB values of a pixel results in a very similar color. Consequently if we classify p as a skin colored pixel then there is a high probability that its neighbors in the color space are skin colored pixels too. This observation is in support of a smaller number of bins for the SPM histogram.

1.3.4 Gaussian classifiers

Gaussian classifiers are parametric skin distribution models with the advantages of being compact, therefore using less memory than SPMs, and able to generalize better using less training data[25].

Single Gaussian

Considering the observations from the thresholding chapter that skin color pixels tend to cluster in a region of the color space we can model that distribution using an elliptical Gaussian joint probability density function, an

example of which was provided by [25]:

$$p(c) = \frac{1}{2\pi^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(c-\mu)^T\Sigma^{-1}(c-\mu)} \tag{1.14}$$

In this equation, c is the color vector, μ the mean vector and Σ the covariance matrix. These can be calculated from the training data as follows:

$$\begin{aligned} \mu &= \frac{1}{n} \cdot \sum_{j=1}^{n} c_j \\ \Sigma &= \frac{1}{n-1} \cdot \sum_{j=1}^{n} (c_j - \mu)(c_j - \mu)^T \end{aligned} \tag{1.15}$$

Here c_j are all the color samples used in training the model. In order to establish that the given color describes skin we can compare p(c) with a threshold that can be determined experimentally for a given training set.

Gaussian Mixture Models

A Gaussian mixture model is a form of unsupervised learning used to identify subpopulations within an overall population, provided they are normally distributed[27].

It has been observed in [28] that a mixture of Gaussians is better suited for skin detection than a single distribution, especially in datasets with multiple illumination conditions.

They represent a generalization of the single Gaussian. A mixture's density function can be calculated as the sum of individual Gaussians[25]:

$$p(c) = \sum_{i=1}^{N} w_i \cdot \frac{1}{(2\pi)^{1/2} \cdot |\Sigma_i|^{1/2}} \cdot e^{-\frac{1}{2}\cdot(c-\mu_i)^T\Sigma_i^{-1}(c-\mu_i)} \tag{1.16}$$

In equation 1.16 c, μ_i and Σ_i are the color vector, mean vector and covariance matrix for the ith Gaussian. Also, each of the N models has a weight, w_i, representing its contribution to the mixture. Determining the unknown parameters can be done with the Expectation Maximization (EM) technique[28]. This is an algorithm of maximum likelihood estimation, or simply speaking finding the parameters that best describe some given data.

The number of Gaussians, N, is also an important aspect. The most common values used in research fall into the 2 to 16 range[25], the idea behind choosing a larger number is to account for various conditions of illumination.

1.4 Skin detection by texture

Chapter 2

Application development

Chapter 3

Conclusions

Bibliography

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

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

[14] 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.

[15] L. G. Shapiro and G. C. Stockman, *Computer Vision*. New Jersey, Prentice-Hall, 2001.

[16] R. Dass, Priyanka, and S. Devi, "Image segmentation techniques," *IJECT*, vol. 3, no. 1, 2012.

[17] C. K. Chow and T. Kaneko, "Automatic boundary detection of the left ventricle from cineangiograms," *Computers and Biomedical Research*, vol. 5, no. 4, pp. 388–410, 1972.

[18] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, 1993.

[19] Y. Nakagawa and A. Rosenfeld, "Some experiments on variable thresholding," *Pattern Recognition*, vol. 11, pp. 191–204, 1979.

[20] N. Senthilkumaran and R. Rajesh, "Edge detection techniques for image segmentation a survey of soft computing approaches," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, 2009.

[21] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," *Computer Vision ECCV 2008*, vol. 5305, 2008.

[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] A. Frich, "Color management guide." `https://www.color-management-guide.com/color-spaces.html/`. Accessed: 2018-04-07.

[24] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," *GRAPHICON03*, pp. 85–92, 2003.

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

[26] J. Yang, W. Lu, and A. Waibel, "Skin-color modeling and adaptation," *ACCV98*.

[27] J. McGonagle, V. Tembo, and A. Chumbley, "Gaussian mixture model." `https://brilliant.org/wiki/gaussian-mixture-model/`. Accessed: 2018-04-09.

[28] M. hsuan Yang and N. Ahuja, "Gaussian mixture model for human skin color and its applications in image and video databases," in *Proceedings of SPIE 99 (San Jose CA)*, pp. 458–466, 1999.