https://doi.org/10.2352/ISSN.2470-1173.2017.4.SRV-352

A Multi-Scale Approach to Skin Pixel Detection

Siddharth Roheda, North Carolina State University, Raleigh, NC, USA, and Hari Kalva, Florida Atlantic University, Boca Raton, FL, USA

Abstract

This paper presents an algorithm to detect skin pixels in an image. Each pixel is classified as a skin or non-skin pixel based on features extracted from its neighborhood. The presented algorithm uses a modified likelihood ratio for classification, and uses a multi-scale approach to classify the pixel in question. The algorithm was developed and evaluated using the ColorFERET dataset. The presented algorithm achieved 95.6 % classification accuracy.

Introduction

Skin detection in images and video has applications in surveilance, analytics, and even video encoder optimization. The ongoing NIST activity of face detection in video also focuses on surveillance application [1]. The goal is to be able to identify pixels in a frame/image that represent the skin of the subject, since it may disclose the identity of the subject. Many applications require the identity of individuals in the image or video to be hidden. Often this is done manually, and the facial area is blurred. This is clearly not a desired approach, not only because it is highly time consuming and requires human interaction, but also because it leaves room for human error. For example, a tattoo on the arm of a person can be used to determine his/her identity, and may be missed by someone who is manually blurring faces in a video. Further, a skin detection approach to identity protection may also be preferred over face detection algorithms, since face detection algorithms would also miss tattoos on parts of the body other than the face.

Another application involving a skin detection task would be one to detect nudity in images/videos. This can be done by detecting skin, and calculating a ratio of skin to non-skin pixels in the image, and further detecting faces to rule out close up selfies.

A successful skin detection may be followed by encryption of the detected pixels, in order to protect identities of individuals in the video, or hide pornographic material.

Related Work

Many methods have been developed and evaluated in the machine learning domain for detection of human skin in images. These methods use features from color models such as HSV and YCbCr and classifiers such as Look-Up Tables, Bayesian, and MLPs. Surveys on the different color models and the methods used for classification are presented in [2], and [3]. A method to detect faces in video combines the use of skin classifier in a YCbCr color space [4] with the Viola-Jones face detection [5]. Face detection is confirmed by the detection of skin in the same area. Gomez and Morales discuss the performance of the Skin Probability Map, and also introduces a new method called Restricted Covering Algorithm (RCA) [6]. RCA searches for candidate rules in parallel, considering two intermixed criteria for selecting new terms.

Dataset

The ColorFERET dataset from NIST [10] has been used for implementation, and performance evaluation of the algorithm. This dataset has about 14,051 images, distributed accross 1,208 faces. There are multiple pose images available for each face. The distribution of images across the dataset can be seen in table 1. Sample images for two of the subjects can be seen in figures 1 and 2.

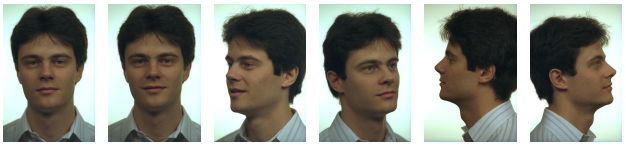

Figure 1. *Sample Images From the ColorFERET Dataset*

Figure 2. *Sample Images From the ColorFERET Dataset*

Initially, a set of 200 skin and non-skin patches were manually extracted, and an SVM classifier was trained using these patches. This classifier was then further used as a seed to extract up to 20,000 skin and non-skin patches from the training set.

The evaluation of the algorithm is done using k-fold cross-validation (k = 10). The target classifier is trained using extracted patches from the training set.

Proposed Method

The algorithm uses Hue and Saturation components from the HSV color space in order to classify pixels into one of the classes, 'skin' or 'non-skin'. Instead of classifying each pixel based on just it's own feature values, we also take into account the neighborhood of the pixel. The contribution of the pixels in the neighborhood is further controlled by weighing their respective features by the squared inverse of their distance from the pixel in question,

Pose Angle (Degrees)	Description	No. in Databse
0	Regular Facial Expression	1962
0	Alternative Facial Expression	1718
0	different illumination	200
+60	Subject Facing to his Left	200
+40	Subject Facing to his Left	200
+25	Subject Facing to his Left	200
+15	Subject Facing to his Left	200
-15	Subject Facing to his Right	200
-25	Subject Facing to his Right	200
-40	Subject Facing to his Right	200
-60	Subject Facing to his Right	200
+22.5	Quarter Left	763
-22.5	Quarter Right	763
+67.5	Half Left	1298
-67.5	Half Right	1246
+90	Profile Left	1342
-90	Profile Right	1398
+45, +10, -10, -45, -80	Random Images	1841

Table 1: Distribution of Face Images in ColorFERET Dataset [10]

as in equations 1 and 2.

$$Hue_M = \frac{(Hue_i + \sum_{j \in N_i} \frac{1}{d_{ij}^2} * Hue_j)}{Z} \quad (1)$$

$$Sat_M = \frac{(Sat_i + \sum_{j \in N_i} \frac{1}{d_{ij}^2} * Sat_j)}{Z} \quad (2)$$

Here, Hue_M and Sat_M are the feature values representing the i^{th} pixel, Hue_i and Sat_i are the Hue and Saturation values for the i^{th} pixel, N_i is the neighborhood of pixel i, and Z is the total number of pixels in N_i.

The distance, $d_i j$, between the i^{th} pixel and it's neighboring pixel j is given by using the distance transform as,

$$d_{ij} = \sqrt{(y_i - y_j)^2 + (z_i - z_j)^2} \quad (3)$$

Where, y_i and z_i are the co-ordinates of the i^{th} pixel and y_j and z_j are the co-ordinates of it's neighboring pixel j.

Feature Selection and Representation

The color of the skin is considered to be a good feature for skin detection, as can be seen from [2], [3], and [4]. From the HSV color space, we select the Hue and Saturation components for representing the skin color. The Value component is rejected, since, it varies over a large range, and is very sensitive to illumination. The rejection of the value component as a feature is further supported by the Forward Feature Selection algorithm.

Forward Feature Selection

The Forward Feature Selection algorithm helps in enhancing generalization, and avoiding over-fitting. In the first iteration, all three (Hue, Saturation, and Value) features are evaluated individually, using the target classifier, and a k-cross validation (k = 10). The feature with the best performance is selected, call it x_1, and added to the array of selected features. In the next iteration, all possible combinations of pairs of features, where the first feature is x_1 and the second feature is selected from the remaining two features are evaluated. The best combination is then added to the array, say $\{x_1, x_2\}$. The performance of the classifier with features $\{x_1, x_2\}$ is compared with the performance of the classifier with x_1, and x_2 is only accepted if the performance with $\{x_1, x_2\}$ is superior. The process continues until an optimum combination is found.

Generalizing, in the i^{th} iteration, the best performing feature, x_i is selected based on the performance with the target classifier. x_i is added to the array of selected features only if the performance of the classifier using $\{x_1, x_2, ..., x_{i-1}, x_i\}$ is better than $\{x_1, x_2, ..., x_{i-1}\}$, and further iterations are performed as long as $i < M$, where M is the total number of features. The selection process is terminated if the performance is found to be inferior.

For our dataset, it is observed that the performance degrades on adding Value to the feature vector, hence, the Forward Feature Selection algorithm selects the subset of features containing Hue and Saturation.

Classification of Pixels

The algorithm for classifying each pixel in an input image/frame is summarized in figure 3. The algorithm begins with a 32x32 size neighborhood, and follows a quad-tree multi-scale approach, until a minimum of 4x4 neighborhood is reached. The 32x32 neighborhood being considered may contain entirely of skin pixels, entirely of non-skin pixels, or may contain a boundary.

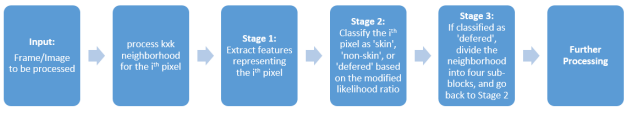

Figure 3. *Flow chart summarizing the algorithm*

The idea is to classify each pixel into 'skin' (w_1) or 'non-skin' (w_2) based on the distribution of its neighborhood. The feature vector, x, is obtained for each neighborhood by calculating the weighted means of hue and saturation using the equations 1 and 2, and the likelihood ratio test, equation 4, is performed to determine which class that neighborhood of pixels belong to.

$$l = \frac{p(x/w_1)}{p(x/w_2)} >_{w_1} (<_{w_2}) 1 \tag{4}$$

Whenever there is a boundary present in the neighborhood, the validity of the decision made using equation 4 is questionable. This effect may lead to a classification bias near boundaries. To address this, likelihood ratio test (equation 4) is replaced by the modified likelihood ratio test from [9], which is represented in equations 5, 6, and 7. This allows us to implement the multi-scale approach for skin pixel detection. If a boundary is detected, the decision is deferred and the 32x32 neighborhood is divided into four sub-neighborhoods, and the modified likelihood ratio test is performed for each of these sub-neighborhoods. This is repeated recursively unless, either a decision is reached, or the window becomes so small that a significant decision cannot be made. In the latter case, the decision is made based on the regions that have been classified.

$$l < a, \text{Classify as skin} \tag{5}$$

$$a < l < b, \text{Defer Decision (possible boundary presence)} \tag{6}$$

$$l > b, \text{Classify as non-skin} \tag{7}$$

Selecting Size of neighborhood

In order to implement the multi-scale approach, the starting size for the neighborhood must be selected. A larger neighborhood provides a more accurate classification over homogeneous regions. Using larger neighborhoods, however, increases the likelihood that the neighborhood contains a boundary. Thus, keeping the neighborhood size as small as possible is also desirable. In this implementation, we use a 32x32 starting size for the neighborhood.

Classifiers

The concept of modified likelihood ratio can be applied to all probabilistic classifiers. We evaluate ad compare some of these classifiers which are further discussed in this section.

Bayesian Classifier

Figure 4 shows the distribution of two classes, considering just one feature. On selecting x_0 as the decision plane, optimum results can be obtained. For this classifier to work, it is required to have prior information. When $p(\mathbf{x}/w1) > p(\mathbf{x}/w2)$, the feature vector, $\mathbf{X}$ most likely lies toward the left of x_0 in figure 1. This means that $\mathbf{x}$ belongs to class 1. In our case, two features have been used, i.e. $\mathbf{x} = [x_1 \ x_2]^T$

We use these distributions along with the modified likelihood ratio in order to determine the membership of the test sample. In cases where the training data is not enough to model a distribution, we make the assumption that the data follows a Gaussian distribution.

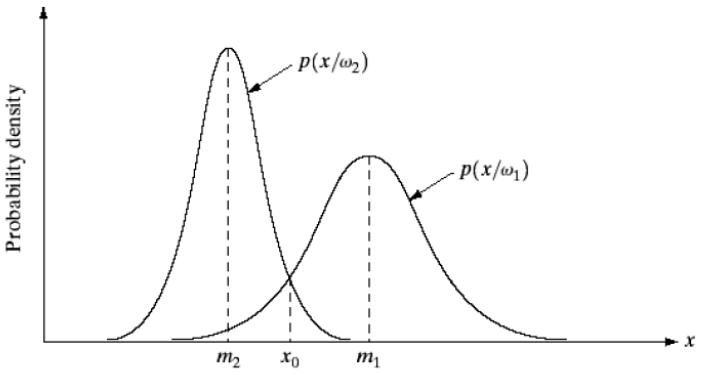

Figure 4. *Bayesian Classifier*

Support Vector Machine

Figure 5 shows a case for separable data, where two possible hyper-planes are shown. Both of these planes correctly separate the data and are viable options for a classifier. But, the dotted plane is a better option, since it is more generalized. SVM selects the most generalized case from all the possible cases.

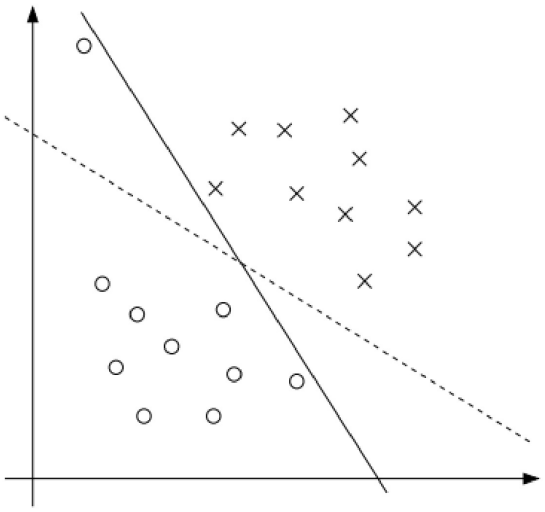

Figure 5. *Possible hyperplanes for separating data*

Traditionally, SVM is trained using the loss function defined in equation 8.

$$min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + \mathbf{C}\sum_{\mathbf{i}} \varepsilon_{\mathbf{i}} \tag{8}$$

$$\text{subject to } y_i(w.x_i) \geq 1 - \varepsilon_i, \ \varepsilon_i \geq 0 \tag{9}$$

Here, $\mathbf{w}$ is the weight vector defining the hyperplane, $\mathbf{y}$ is the vector of labels of the training samples, ε is the vector of slack variables, and C is a constant that controls the relative influence of the two competing terms.

In case of non-linear data, a kernel functional must be used to transform the features into a linear space. Data in this implementation, as is the case with most real world data, is non-linear. We

use the radial basis function to transform the data into a higher dimensional space, where the data is linearly separable. The radial basis function is defined as in equation 10.

$$K(x,z) = e^{-\frac{||x-z||^2}{2\sigma^2}} \tag{10}$$

In order to use the modified likelihood ratio with the Support Vector Machine, the score of an incoming sample, x, $f(x) = w.x$, must be converted into a probability. This is done by using the formulation provided by Platt [11]. In the paper, he suggests using a parametric model to fit the posterior probability P(y = 1—f). The form of the parametric formula is given as,

$$P(y = 1|f) = \frac{1}{1+exp(Af+B)} \tag{11}$$

Here, A and B are the parameters to be estimated, and are fit using the maximum likelihood estimation from the training set, (f_i, y_i). Now, define a new training set, (F_i, t_i), where t_i are the target probabilities, as defined in equation 12 [11].

$$t_i = \frac{y_i+1}{2}, \quad y_i = \pm 1 \tag{12}$$

Following this, the negative log likelihoodof the training data is minimized, as in equation 13 [11].

$$min\{-\sum_i t_i log(p_i) + (1-t_i) log(1-p_i)\} \tag{13}$$

Where,

$$p_i = \frac{1}{1+exp(Af_i+B)} \tag{14}$$

This allows us to implement the multi-scale approach using a modified likelihood ratio, with Support Vector Machine as the underlying classifier.

k-Nearest Neighbors

For an unknown feature vector **x**, and some distance measure, the k-Nearest Neighbor rule [12] is summarized as follows:

- Out of the N training vectors, identify the k nearest neighbors, regardless of the class label, using the selected distance measure.k is usually chosen such that it is not a multiple of the number of classes, M.
- Out of these k samples, identify the number of vectors, k_i, that belong to the class w_i, $i = 1,2,..,M$. Here, $\sum_i k_i = k$.
- Assign **x** to the class w_i with the maximum number k_i of samples.

The probability of class membership for the test sample can be obtained by using the following equation,

$$P(w_i|x) = \frac{\sum_{k_i \in \omega_i} k_i}{k} \tag{15}$$

These probabilities are then combined with the multi-scale approach using modified likelihood ratio, allowing us to use k-NN as the underlying classifier.

Results and Conclusion

The performance of the classification task is evaluated and validated on the ColorFERET face dataset provided by NIST [10]. k-fold cross-validation with k=10 is performed over the entire dataset, and the performance is reported in table 2. The modified likelihood ratio can be extended to other probabilistic classifiers as discussed in the section on classifiers, and we compare the performance for Bayesian, Support Vector Machine, and k-Nearest Neighbors Classifiers. The accuracy of the discussed approach is seen to be superior to approaches using single pixel based features for classification in [3], where, the maximum accuracy achieved is 89.84 %. The processing times in table 2 are with respect to images of size 512x768. Examples of successfully detected skin pixels can be seen in figures 6, 7, 8 and 9.xThese images cover successful skin detection over several skin tones, hence confirming the robustness of this algorithm over different skin colors.

The performance suffers when the subject is wearing clothes very close to skin color. Such a case can be seen in figure 10. This is expected since the classifiers are trained with features describing the color of the skin. This could be solved by adding another feature, such as texture, in order to be able to differentiate between skin and clothes. This issue is not addressed in this paper, and is something that should be explored in the future.

Figure 6. *Original Image on the Left, Processed Image on the Right*

Figure 7. *Original Image on the Left, Processed Image on the Right*

Classifier type	Classification Accuracy	Processing Time
SVM	94.2 %	20.54 secs
kNN (k=10)	92.3 %	7.30 secs
Bayesian	95.6 %	5.33 secs

Table 2: Classification Accuracy for Multi-Scale Approach using a large training set (about 20,000 skin and non-skin patches) is used

While, table 2 uses a much larger training set (20,000 patches), we also evaluate the performances of these classifiers when a smaller training set is used (200 patches) in table 3.

Classifier type	Classification Accuracy	Processing Time
SVM	92.6 %	12.54 secs
kNN (k=1)	92.9 %	3.22 secs
Bayesian	86.8 %	2.66 secs

Table 3: Classification Accuracy for Multi-Scale Approach using a small training set (about 200 skin and non-skin patches) is used

As can be seen from table 2 and 3, the performance is best when a Bayesian Classifier is used with a larger training set, but it significantly degrades when the training set is smaller, since the approximation for the distribution of the skin and non-skin classes is not good. Further, Fine k-NN (k=1) does a great job when a smaller training set is available. Selection of the underlying classifier is hence dependent on amount of training data available, and also on processing time constraints. One would prefer Bayesian Classifier if a larger dataset is available, but stick with Fine k-NN if the dataset is small.

Figure 8. *Original Image on the Left, Processed Image on the Right*

Figure 9. *Original Image on the Left, Processed Image on the Right*

Figure 10. *Original Image on the Left, Processed Image on the Right (A failed detection, due to the shirt being of the same color as skin)*

References

[1] N. US Department of Commerce, Face in Video Evaluation (FIVE). [Online]. Available: http://www.nist.gov/itl/iad/ig/five.cfm. [Accessed: 15-Aug-2016].

[2] V. Vezhnevets, V. Sazonov and A. Andreeva , "A Survey on Pixel-Based Skin Color Detection Techniques," Proc. Graphicon, 2003.

[3] S. L. Phung, A. Bouzerdoum and D. Chai, "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison," Skin Segmentation Using Color Pixel Classification: Analysis and Comparison, 2005.

[4] C. Liensberger, J. Stottinger and M. Kampel, "Color-Based Skin Detection and its Application in Video Annotation," Computer Vision Winter Workshop , 2009.

[5] P. Viola and M. Jones, "Robust Real-Time Face Detection," International Journal of Computer Vision, 2004.

[6] G. Gomez and E. Morales, "Automatic feature construction and a simple rule induction algorithm for skin detection," ICML workshop on Machine Learning, 2002.

[7] G. Gomez and E. Morales, "Automatic feature construction and a simple rule induction algorithm for skin detection," ICML workshop on Machine Learning, 2002.

[8] A. Albiol, L. Torres and E. Delp, "OPTIMUM COLOR SPACES FOR SKIN DETECTION," ICIP, 2001.

[9] C. H. Fosgate, H. Krim, W. W. Irving, W. C. Karl and A. S. Willsky, "Multiscale Segmentation and Anomaly Enhancement of SAR Imagery," IEEE TRANSACTIONS ON IMAGE PROCESSING, 1997.

[10] NIST, Information Technology Laboratory/Information Access. [Online] Available: https://www.nist.gov/itl/iad/image-group/color-feret-database [Accessed: 15-Aug-2016]

[11] John C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods", Advances in large margin classifiers, 1999.

[12] S. Theodoridis and K. Koutroumbas, Pattern Recognition.

Author Biography

Siddharth Roheda

Siddharth Roheda received his B.Tech. in Electronics and Communication from Nirma University, India (2015), and is currently pursuing his PhD at the North Carolina State University.

Hari Kalva

Hari Kalva is a Professor, Associate Chair, and the Director of the Multimedia Lab in the Department of Computer & Electrical Engineering and Computer Science at Florida Atlantic University (FAU). Dr. Kalva has over 20 years of experience in multimedia research, development, and standardization. He has made key contributions to technologies that are now part of MPEG-4 standards. His current research focuses on understanding and applying human visual perception, cognition, and social context to optimize visual information processing. Dr. Kalva received a Ph.D. and an M.Phil. in Electrical Engineering from Columbia University in 2000 and 1999 respectively. He received an M.S. in Computer Engineering from Florida Atlantic University in 1994, and a B. Tech. in Electronics and Communications Engineering from N.B.K.R. Institute of Science and Technology, S.V. University, Tirupati, India in 1991.