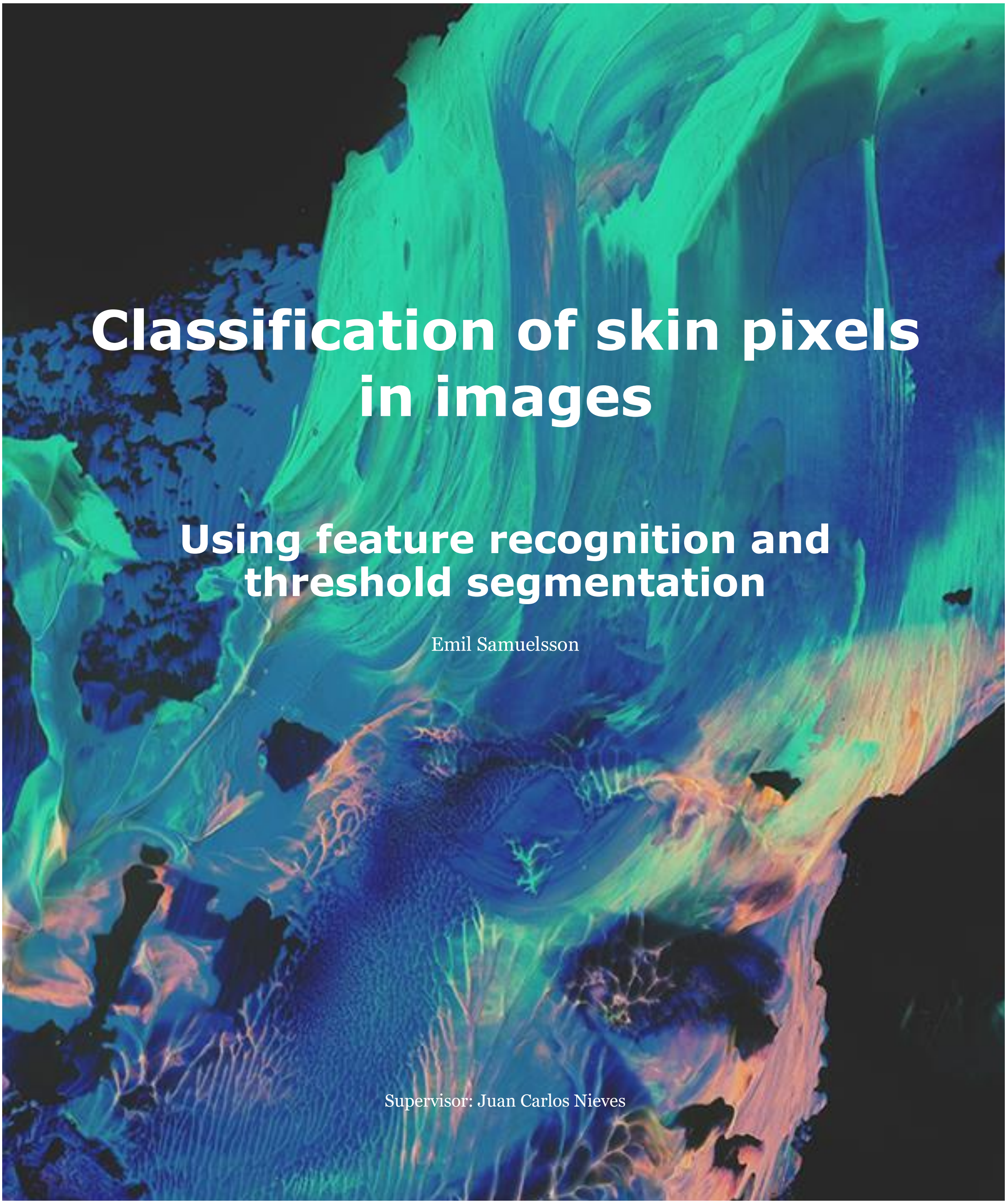
UMEÅ UNIVERSITY



Classification of skin pixels in images

Using feature recognition and threshold segmentation

Emil Samuelsson

Supervisor: Juan Carlos Nieves

Bachelour thesis, 15 hp
Kandidatprogrammet I datavetenskap, 180 hp
Spring term 2018

Abstract

The purpose of this report is to investigate and answer the research question: *How can current skin segmentation thresholding methods be improved in terms of precision, accuracy, and efficiency by using feature recognition, pre- and post-processing?*

In this work, a novel algorithm is presented for classification of skin pixels in images. Different pre-processing methods were evaluated to improve the overall performance of the algorithm. Mainly, the methods of image smoothing, and histogram equalization were tested. Using a Gaussian kernel and contrast limited adaptive histogram equalization (CLAHE) was found to give the best result. A face recognition technique based on learned face features were used to identify a skin color range for each image. Threshold segmentation was then used, based on the obtained skin color range, to extract a skin map for each image. The skin maps were improved by testing a morphology method called closing and by using contour detection for an elimination of large false skin structures within skin regions. The skin maps were then evaluated by calculating the precision, recall, accuracy, and f-measure using a ground truth dataset called Pratheepan. This novel approach was compared to previous work in the field and obtained a considerable higher result. Thus, the algorithm is an improvement compared to previous work within the field.

Introduction

Skin segmentation is a useful tool with many applications where examples are hand gesture recognition [1], face recognition [2], explicit image detection [3], tracking in real-time video [4], and facial expression analysis [5]. Currently, skin segmentation has many limitations. Solving these could allow this technique to become a leading method for the mentioned applications due to the simplicity of the method in terms of implementation.

The limitations of the skin segmentation technique are because they focus on the color and illumination of pixels. This directly implies certain problems that must be solved. One is the large variety of ethnicities with a wide range of skin color, another being uneven illumination, and distinguishing between skin and skin colored objects.

In this report, pre- and post-processing, skin segmentation thresholding, and feature recognition is combined into an algorithm for skin segmentation. The algorithm is compared to previous work based on the Pratheepan Dataset and evaluated using precision, accuracy, and f-measure.

In the following sections, the methods used will be briefly explained in the method section, followed by a result section that shows all results obtained, and lastly, the paper will be finished with a discussion of the results, how the introduced research question was answered, future work, and limitations.

Related work

Previous work has tried to solve the limitations of skin segmentation. In the work of Kolkur, et al [6] they implemented a thresholding algorithm based on three color spaces, RGB (Red, Green, Blue), HSV (Hue, Saturation, Value) and YCbCr (Luminance, Chrominance). They found that using thresholds in these color-spaces gave high precision and accuracy in non-complex images.

Limitations of this work are using it for real-time analysis and for complex images that contain objects like the skin pixels in the background.

Another approach from the work of Platzer, et al [3] they implement a skin segmentation algorithm that can be used for recognition of explicit image detection. Their work mainly focused on trying different pre- and post-processing steps in combination with learning a color threshold for skin pixels. They found that using an auto contrast function to equalize the contrast in the image as pre-processing and a morphology closing function as post-processing gives good results. However, as they state, their skin detection is limited because it does not work with skin colors that are too bright or too dark.

In the work of Erdem, et al [7], they develop a face detection algorithm by combining feature recognition and a Bayesian classifier for skin detection. This work shows that by combining two methods they can achieve high precision for face detection.

Aim

Against this background, the aim of this work is to answer the following question: *How can current skin segmentation thresholding methods be improved for in terms of precision, accuracy, and efficiency by using feature recognition, pre- and post-processing?*

Theory

In this section, the theory of concepts used in the method section will be briefly explained. These concepts are the ground truth evaluation, image smoothing, histogram equalization, Color spaces, features recognition using haar-like features, and the morphology method of closing.

Image smoothing

Image smoothing is a technique for evening out pixel values in an image by applying a kernel function. By smoothing the image, noise and edges get evened out which can allow better analysis of the image. Three separate common kernels for this are the average, median, and Gaussian kernels. The average kernel replaces the current pixel value by the average pixel value from the surrounding pixels. The median kernel works in a similar way to the average kernel, but instead of the average pixel value, it computes the median. The Gaussian kernel is a computationally expensive kernel that uses the mathematical function seen in equation (1) to compute a new value where the pixels that are closest is weighted more than pixels further away.

$$(1)\ f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma^2}}$$

$$(2)\ \sigma = standard\ deviation$$

$$(3)\ \mu = expected\ value$$

Histogram equalization

Another problem in image analysis is uneven contrast in images. This is especially a problem in skin segmentation because the uneven contrast in the image creates a larger color range that needs to be segmented. Two equalization techniques are histogram equalization (HE) and contrast limited adaptive histogram equalization (CLAHE) [8]. Both techniques goal is to improve the contrast by using the most common contrast value and applying them to the images. The main difference between the techniques is that HE uses the complete image histogram to equalize on a global basis, while CLAHE is applied on local regions with the histograms for that region.

Color spaces

A color space is a representation of an image. There exist many different color spaces based on different color theories. Here the RGB, YCbCr, and CIELAB color spaces are briefly explained.

RGB

This is a standard image representation that uses three separate channels "Red", "Green" and "Blue". In the image, each pixel has one value for each color channel. These values are in the range 0-255. These values together represent the actual color of the pixel.

YCbCr

This is a color space with three channels, Y, Cb, and Cr. The Y component is the luminance, ergo the light intensity of the pixel. Cb is the difference between the Y and blue components and Cr is the difference between Y and red components.

CIELAB

The CIELAB color space is based on perceptual uniformity and contains three components. These components are the L (Lightness), a* (green-red), and b* (blue-yellow). This color space represents the how two colors differ when observed with the human eye.

Feature recognition using haar-like features

Haar-like feature recognition is a method for identification of features in images. The principle of this approach is to divide the image into rectangles then divide each rectangle in half. In each half of the rectangle, the sum of all pixel values is computed. Then the halves are subtracted to get the difference between them. This difference is used to train and learn features for recognition such as for face features [9].

Morphology method of closing

Closing is based on two principles followed by each other. The first is erosion where a kernel function is applied to the image. If the surrounding pixels in the kernel all is white, the pixel is set to white, else to black. This process eliminates black pixels and makes objects smaller due to pixels near the edges is set to black. To reverse the effect of a smaller object the principle of dilation is applied after erosion. Dilation is the reverse of erosion, it sets pixels to white if any pixel in the kernel is white. Thus, erosion followed by dilation removes noise in objects in the skin mask. This process is called closing. The effect of erosion, dilation, and closing are seen in Figure 1.

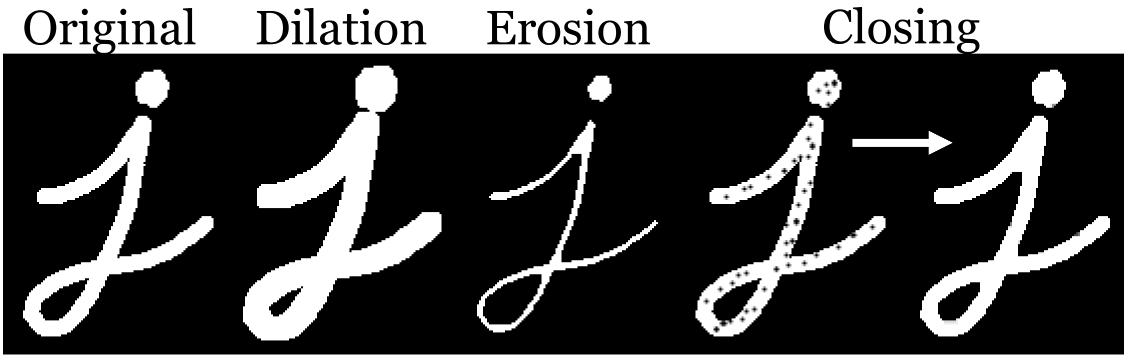


Figure 1. The effects of Dilation, Erosion, and Closing.

Ground truth evaluation

A way to evaluate skin segmentation is by using a ground truth dataset, containing images where the skin pixels have been manually selected. The ground truth is essentially four values that are obtained by comparing the ground truth image and the obtained skin map. True positives – pixels that are correctly classified as skin pixels, True negative – pixels that have been correctly been assigned as non-skin, false positive – pixels that are classified falsely as skin pixel, and false negatives – skin pixels that have been classified as non-skin pixels. An example can be seen in Figure 2, where the left picture is the original image and the right is the manually selected skin map for the ground truth.

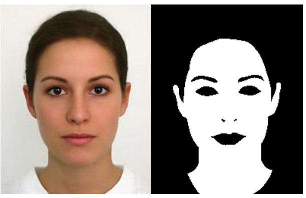


Figure 2. An example of a manually selected skin pixel map. To the left is the original image to the right is the manually selected ground truth skin map. The image is from the Pratheepan dataset.

The ground truth values can be used to calculate the precision, the recall, accuracy, and F-measure. Precision measures the percentage of correctly identified skin pixels out of all identified skin pixels, its calculated by using equation 1.

$$(1)\ \ Precision = \frac{true\ positive}{true\ positive + false\ positive}$$

Recall measures the percentage of correctly classified skin pixels out of the total skin pixels, recall is calculated like precision as seen in equation 2.

$$(2)\ \ Recall = \frac{true\ positive}{true\ positive + false\ negative}$$

Accuracy is the percentage of pixels that were correctly classified. It can be calculated by dividing the sum of true positives and true negatives by the total number of pixels (see equation 3).

$$(3)\ \ Recall = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

F-measure combined recall and precision into a metric that can be used to compare different models. This is useful when you want a model with both high precision and recall. F-measure is calculated by equation 4.

$$(4)\ f - measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Method

In this section, the workflow for the developed algorithm is described by introducing each method in the order they are used. This includes the dataset, the pre-processing methods, face detection, the skin color range selection, the post-processing methods, and the evaluation using the ground truth.

Dataset

For evaluation of the developed algorithms, the Pratheepan Dataset was used [10]. The data set is randomly selected face and family photos from google. It contains 32 face and 46 family photos. The ground truth has been selected for each image, i.e. skin pixels has been manually selected in each image to obtain the ground truth. The dataset contains images of people with a wide range of ethnicities.

Preprocessing of images

In image analysis, an important step is preprocessing. One of the uses of preprocessing is to normalize images. However, preprocessing can be used for different means depending on what the end goal is. In this study, a combination of pre- and post-processing techniques has been tested to improve classification of skin pixels.

The first pre-processing step was to do image smoothing to remove noise in the images. The three most common kernels were tested for this purpose. These kernels were the average, median, and Gaussian kernels. They were evaluated with the final algorithm to see their impact on the classification of skin pixels.

To normalize the contrast in the images two techniques were tested. The first technique was histogram equalization (HE) and the second was contrast limited adaptive histogram equalization (CLAHE) [8]. To use these techniques the images were converted to the CIELAB color space and then the techniques were applied to the lightness component L. The effect of the two techniques was evaluated with the final algorithm.

Skin color segmentation

Skin color segmentation is an approach where skin pixels are segmented out from the images by using thresholds in one or more color spaces. In this work, the segmentation was based on the YCbCr color space.

To be able to identify the skin color range of skin in pictures, a face detection approach was chosen. The face detection algorithm was based on machine learning approach using Haar-like features [9]. A pre-trained model from OpenCV [11] for face detection was used to identify faces in each image.

After identification of faces, a square of 20x20 pixels was selected in the middle of each identified face. These squares were used to compute threshold values in the YCbCr color space for that person's skin color. The max and min values of all the color space components (Y, Cb, Cr) were selected as the range of skin pixels to use for skin pixel segmentation from the images. The selection of the skin color range is visualized in Figure 3.

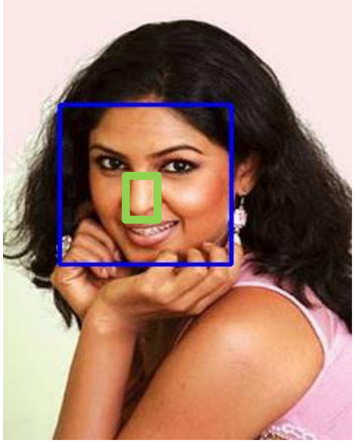


Figure 3. Identification of a person's face (blue square) and selection of a skin color range for skin classification (green square).

The min and max components for the Y, Cb, and Cr components that were selected in the image were then used to extract the pixels in the image that fall within the min and max range. To do this efficiently, a Cython [12] module was implemented in python [13] to loop through all pixels and select those in range. The pixels within the range where set to white color, the rest as black. Thus, a skin mask is created.

Post-processing of images

The skin mask created from the images can contain noise in form of black pixels within skin structures. Post-processing can help to limit this form of noise in the skin mask. Mainly two post-processing methods were tested, morphology closing and contour detection.

The method of closing was used to remove small noise. However, another method is needed to remove larger noise from large structures such as the face region. The tested approach was to use contour detection to identify the edges of all objects. The objects were then ordered in a hierarchical tree where objects within other objects are classified as the larger objects child. Thus, smaller objects can be removed by pruning the hierarchical tree. Both post-processing methods were tested independently and together with the developed algorithm to evaluate their impact on the classification.

Evaluation

The different pre-processing and post-processing were evaluated using the ground truth images from the data set. The ground truth values, true positive, true negative, false positive, and false negative were used to calculate the precision, recall, accuracy, and f-measure. These values were used to compare the different processing methods effect on the final model.

Final model

By testing different pre- and post-processing methods with the haar-like feature detection a final model could be derived. A Gaussian kernel together with CLAHE was used as pre-processing for the face detection using Haar-like features for identifying a color range for segmenting skin pixels from the images. Post-processing was done by using the method of "closing" and hierarchical contour elimination. The pseudo code for the final algorithm can be seen in Figure 4.

```
function SkinPixelExtraction(frame)
GaussianBlur(frame)
CLAHE(frame)
faces <- FaceDetection(frame)
colorRange <- getColorRange(frame, faces)
for all pixel in frame:
    if pixel in colorRange:
        frame[pixel] <- white
    else:
        frame[pixel] <- black
Closing(frame)
ExtractContours(frame)
return frame
```

Figure 4. Pseudo code for the developed algorithm.

Result

An algorithm was developed for extracting skin pixels from images. The model was based on face detection using Haar-like features to find thresholds for the YCbCr color space. These thresholds were used to extract skin pixels from the image. Different pre- and post-processing methods where tested to improve the model. A dataset called the Pratheepan Dataset were used. Only the face images from dataset were used when testing the different methods.

Three different smoothing kernels were tested for pre-processing the images. An average, median, and a Gaussian kernel. The precision, recall, accuracy, and F-measure was computed using each of the kernels. No kernel was also tested. The result can be seen in Table 1. The Gaussian kernel achieved higher values compared to the rest.

Table 1. Result for testing different kernels with the final algorithm. The kernels tested were Gaussian, median, and average kernels.

Kernel	Precision	Recall	Accuracy	F-measure
No kernel	0.9034	0.8277	0.7843	0.8270
Gaussian	0.9173	0.8476	0.8150	0.8594
Median	0.8697	0.8121	0.7717	0.8191
Average	0.8660	0.8399	0.7843	0.8313

The effect of each kernel was visualized on one of the images in the dataset. The Gaussian kernel created a more even blur in the image comparing to the other kernels, as illustrated in Figure 5.

Original Gaussian kernel Median Average kernel

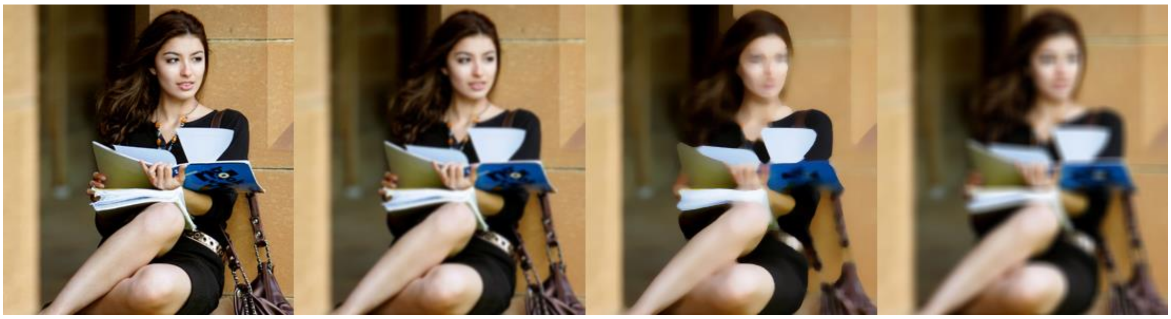


Figure 5. The effects of using no, Gaussian, median, and average kernel respectively. The image is from the Pratheepan Dataset.

To normalize the contrast in the images, two equalization techniques were tested, contrast limited adaptive histogram equalization (CLAHE) and histogram equalization (HE). Using no equalization method were also tested. The result in terms of precision, recall, accuracy, and f-measure can be seen in Table 2. CLAHE obtained the highest values.

Table 2. The result from testing both the HE and CLAHE equalization methods with the developed algorithm.

Equalization	**Precision**	**Recall**	**Accuracy**	**F-measure**
CLAHE	0.9173	0.8476	0.8150	0.8594
HE	0.8920	0.7896	0.7662	0.8020
None	0.8866	0.8056	0.7728	0.8099

The effects of each equalization were visualized on an image from the Pratheepan Dataset, as seen in Figure 6. CLAHE (middle image) evened out the contrast more compared to HE (right image).

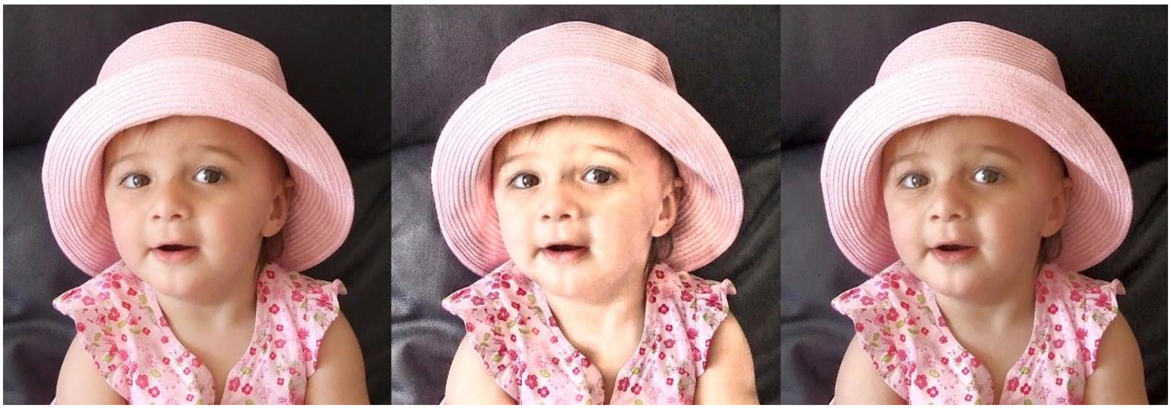


Figure 6. The effect of equalizing the contrast on an image selected from the Pratheepan dataset. The Original image to the left, CLAHE in the middle, and histogram equalization to the right.

For post-processing, two separate methods were tested in combination and separately. The first method was morphology method of closing while the second method was contour detection and elimination of small objects within others. The results for this can be seen in Table 3. Using both methods gave the highest values.

Table 3. The result from the post-processing methods combined with the developed algorithm. Precision, recall, accuracy, and f-measure for each method.

Post-processing	Precision	Recall	Accuracy	F-measure
Closing & Contour	0.9173	0.8476	0.8150	0.8595
None	0.8676	0.8961	0.8113	0.8672
Closing	0.8931	0.8712	0.8149	0.8647
Contour	0.8930	0.8733	0.8136	0.8637

The final model was validated by testing on both the face and family subsets but also on the full set of the Pratheepan dataset. The result can be seen in Table 4. Using only the face subset gave the highest values.

Table 4. The result for the final model obtained from both subsets separately and together.

Dataset	Precision	Recall	Accuracy	F-measure
Face subset	0.9173	0.8476	0.8150	0.8595
Family subset	0.9053	0.5660	0.5972	0.6545
Both	0.9098	0.6840	0.6893	0.7418

Further analysis was made to investigate what kind of images the algorithm did not work well with. The f-measure values for each image were plotted to check how many images that got low and high f-measure values. In Figure 7, three values have zero f-measure, and in total 20 images got below 0.6 f-measure.

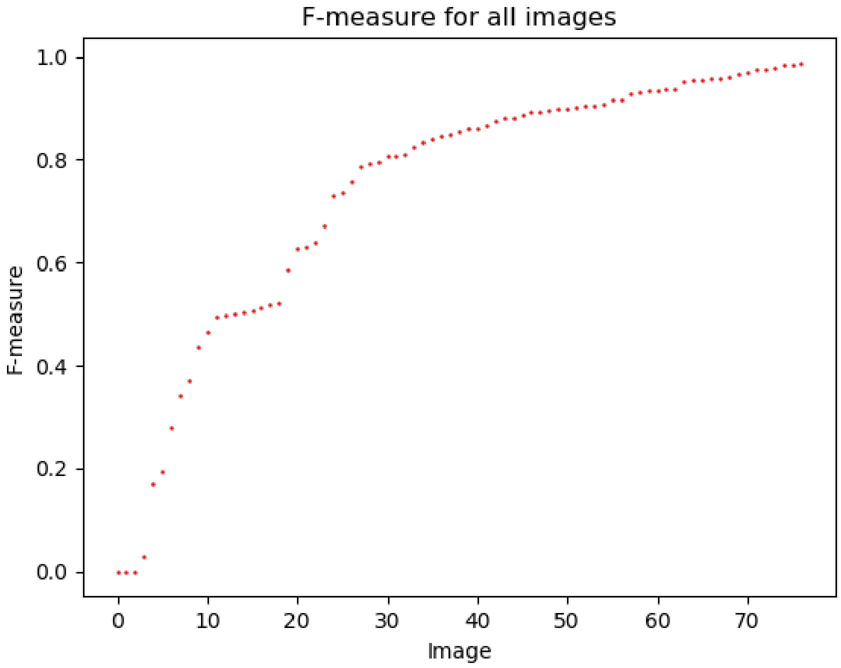


Figure 7. Plotted f-measure values for all images in the Pratheepan dataset.

To check what kind of images that got low f-measure a similar plot was made for both the face family subsets. The plot for the family subset can be seen in Figure 8. All the images below 0.5 f-measure were present in the family subset.

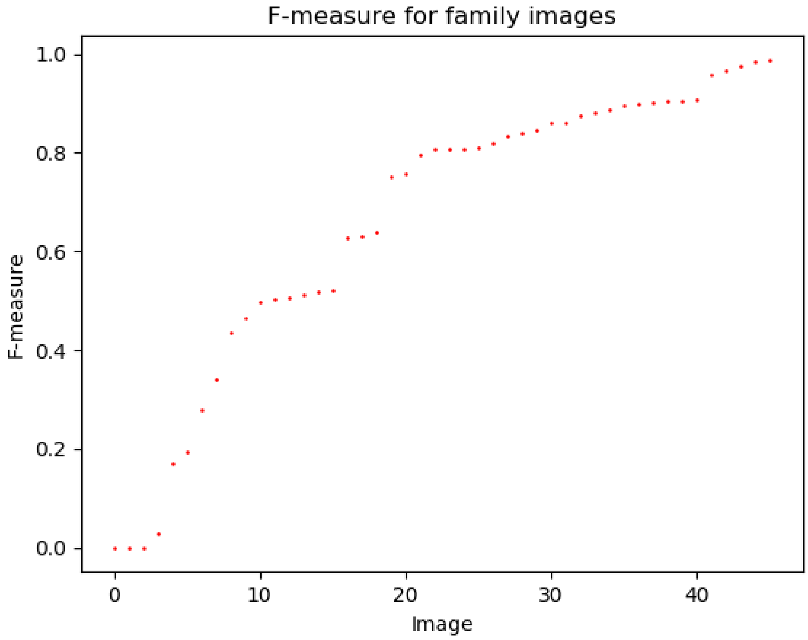


Figure 8. Plotted f-measure values for the family subset of the Pratheepan dataset.

In figure 9, the f-measure values for the face subset are shown. Majority of the images have f-measure above 0,85, with only a few values in the range 0.5 to 0.85.

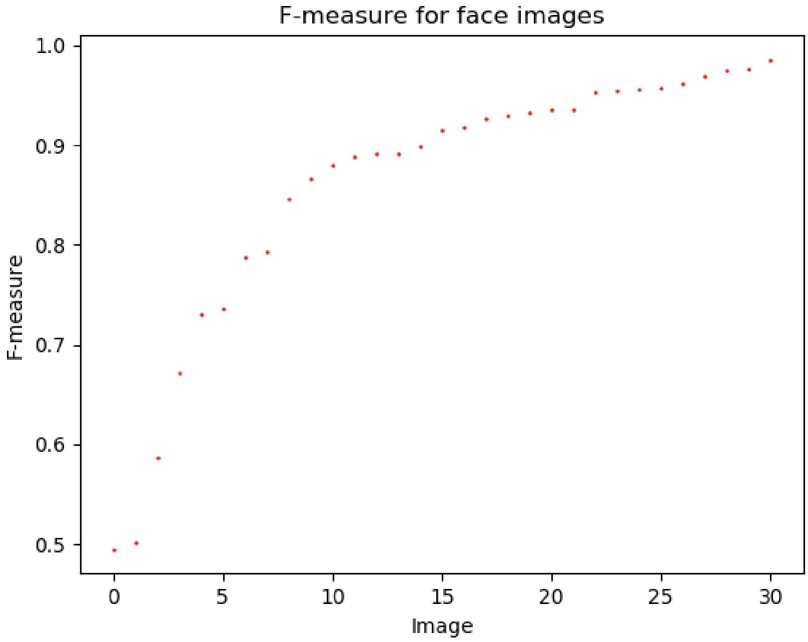


Figure 9. Plotted f-measure values for the face subset in the Pratheepan dataset. Many of the images had greater than 0,85 f-measure.

The skin maps from an image with high and low f-measures were compared. The image shown in Figure 10 contains the original image to the left, the extracted ground truth in the middle, and the ground truth to the right. This image is considered a non-complex image. The extracted skin map is very similar to the ground truth. The f-measure for this image was 0.975.

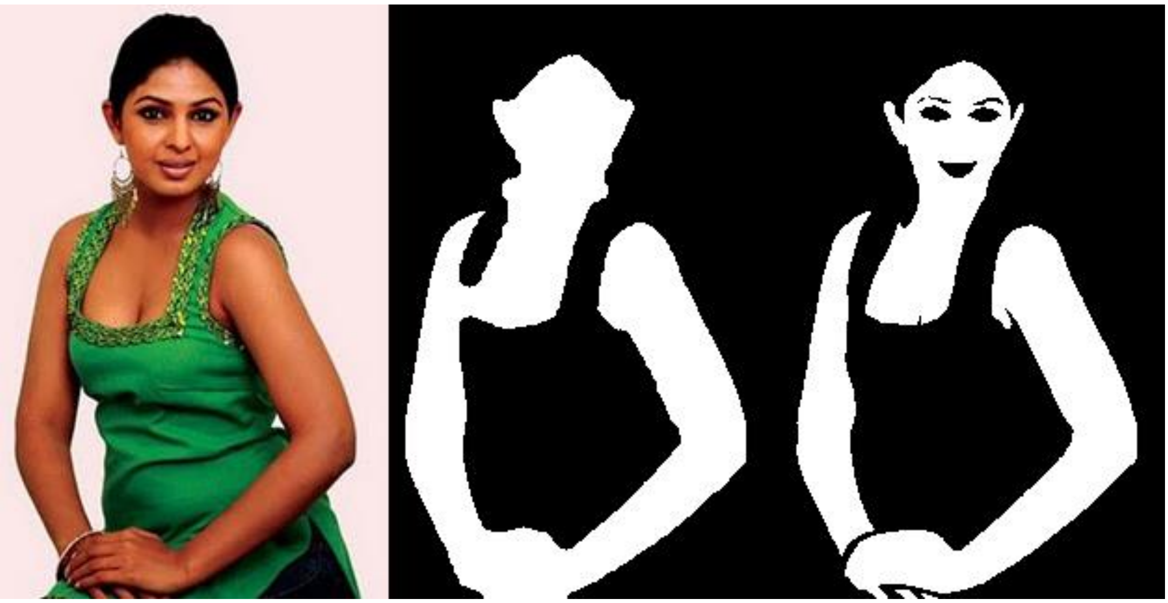


Figure 10. Illustration of obtained skin map from the final model. The original picture to the left, obtained skin map in the middle, and the ground truth image to the right. The obtained skin map is similar to the ground truth.

The image with a poor extracted skin map can be seen in Figure 11. The f-measure value was 0.58. This image is considered a complex image due to many objects in the background but also because of the color of some objects are like the skin pixels.

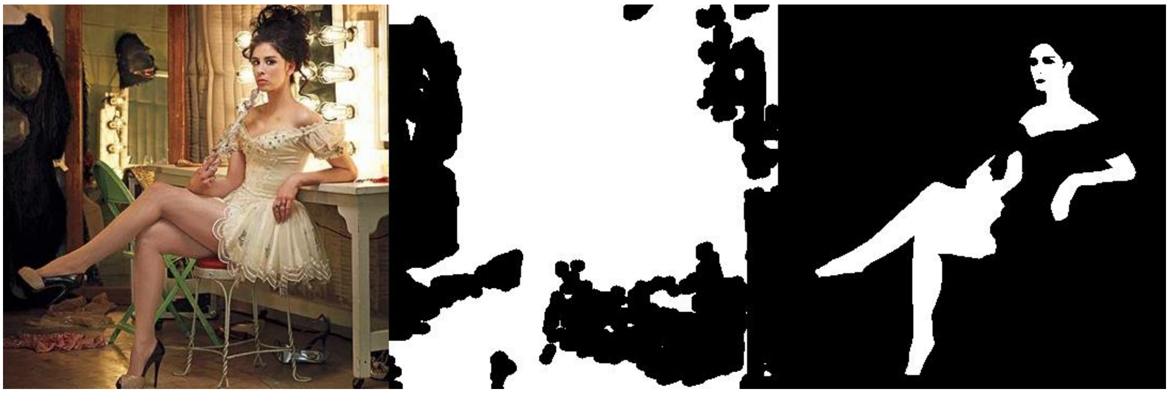


Figure 11. Illustration of obtained skin map from the final model. The Original image, the obtained skin map, and the ground truth skin map are shown respectively. The image is considered complex due to many skin-colored objects in the background. The obtained skin map does not match the ground truth.

Discussion

To aim of this report was to answer the research question: *How can current skin segmentation thresholding methods be improved for real-time applications in term of precision, accuracy, and efficiency by using feature detection, pre- and post-processing?*

An algorithm for classifying skin pixels in images was developed based on face detection and threshold segmentation. The face detection was based on Haar-like features. Pre- and post-processing was applied to normalize and equalize the images. The processing methods were tested in combination with the algorithm to see if the model performance could be improved.

To see if image smoothing could improve the result three smoothing kernels were tested. The average, median, and Gaussian kernels. The result indicates that using no kernel is better than using the average or median kernels. However, the Gaussian kernel improves the precision, recall, accuracy, and f-measure compared to using no kernel. Thus, the final algorithm uses the Gaussian kernel as a pre-processing step.

Equalization of contrast in images is an important pre-processing step because images are often unevenly illuminated. Equalizing the contrast could thus improve the general performance of the algorithm. Histogram equalization and contrast limited adaptive histogram equalization (CLAHE) are two different equalization methods. CLAHE was found to improve the model more than HE in terms of precision, recall, accuracy, and f-measure. The result could also be observed in Figure 6, where the middle picture which represents the CLAHE clearly evened out the contrast.

The main algorithm for classification of skin pixels was based on Haar-like features to identify the faces. After faces have been found the color range of the faces were selected to obtain a threshold that can be used for segmenting out all skin pixels in the image. This approach as seen in the result sections performs well with about 85% f-measure. However, this method has drawbacks. In the family subset of the Pratheepan, no faces were able to be found in two of the faces due to the resolution of the image. This makes the algorithm unusable on those images.

Currently, the face recognition algorithm uses a pre-trained machine learning model for the Haar-like features. Implementing an optimized model for the current problem could improve the results. Another implementation that is needed for the face detection is segmentation based on all detected faces because currently it only selects the first best face found. By adding the suggested improvements, the model could obtain even higher result for images where there are many people with different skin color.

The skin threshold selection in the algorithm is currently based on selecting a range of skin pixels from a square in the middle of the face. This is a rough technique were parts of eyeshadows and similar could be used as values. This is a problem in some images and could also be one of the reasons that the skin maps for the complex images got low f-measure. An improvement of this selection would be to ensure a specific region like the nose got selected. This could be done with either using Haar-like features for nose detection or by using contours.

After skin segmentation, images can still contain noise and false structures within skin objects. These can be eliminated by using morphology and by using contour detection to remove inner structures. The images were post-processed with these two methods, the morphology method “closing” and hierarchical contour elimination. Using the methods separately gave the same f-measure values which indicate that they are equally good. However, it was found that combining these yield better results. Even so, elimination of objects by hierarchical contours can be a problem where the background is identified as skin because the model may instead remove the wanted skin pixels if contained within a larger structure.

Analyzing the results of the algorithm, it performs well with the face dataset, but it is not perfect for complex images. The approach itself seems promising but it needs to be further improved. However, future work depends on what the end goal is. If the goals are to extract face and hand gestures for hand recognition, it could be extended to use haar-like feature detection of hands, while for facial expression recognition a better trained haar-like feature model would suffice.

Previous work

In the work of Platzer [3] and Kolkur [6], they developed skin threshold approaches based on learning threshold values with machine learning. These approaches work well but it does not circumvent the problem with complex images where skin like objects is present in the background. This is likely because the models need to be generalized for all ethnicities. Using Haar-like features for identification of skin color range for threshold segmentation allows the thresholds to be very specific for the identified person. Thus, the risk for identification of false skin pixels is lower.

In the article for the threshold segmentation by Kolkur et al [6], they got precision of 0.85 for six selected images from the Pratheepan dataset. Of these images, only one were from the family subset, and none were complex images. The algorithm presented in this report got a precision of 0.91 for the family subset containing all complex images. Thus, the algorithm is an improvement compared the work of Kolkur et al.

A K-means clustering method for classification of skin pixels was proposed by Buza, et al. [14]. This approach used clustering to group up and classify skin pixels, the work was evaluated on the Pratheepan dataset. They obtained f-measure values for the face subset of 0.74, for the family dataset 0.39, and for the total set 0.57. These values are considerably lower than the f-measure of 0.86 obtained by the presented algorithm in this report.

Conclusion

An algorithm was developed using different pre- and post-processing methods, and Haar-like feature detection. The approach obtained a better result than previous work in the field and show promising results that could be further improved by tuning the parameters in the processing methods and by creating a customized haar-like feature detection model.

References

[1] L. Y. Duan Hong, "A Method of Gesture Segmentation Based on Skin Color and Background," in *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*, Los angeles, 2013.

[2] A. A. Mohammadreza Hajiarbabi, "Face Detection in Color Images Using Skin Segmentation," *Journal of Automation, Mobile Robotics & Intelligent Systems,* vol. 8, 2014.

[3] C. Platzer and a. et, "Skin Sheriff: A Machine Learning Solution for Detecting," in *Proceedings of the 2nd international workshop on Security and forensics in communication systems*, Tokyo, 2014.

[4] S. Bilal, R. Akmeliawati, M. J. E. Salami and A. A. Shafie, "Dynamic approach for real-time skin detection," *J Real-Time Image Proc,* 2012.

[5] K. P. Bourbakis N., "Skin-based Face Detection-Extraction and Recognition of Facial Expressions," *Applied Pattern Recognition. Studies in Computational Intelligence,* vol. 91, 2008.

[6] S. Kolkur and a. et, "Human Skin Detection Using RGB, HSV and YCbCr Color," *Atlantis press,* pp. 324-332, 2017.

[7] C. Erdem, S. Ulukaya, A. Karaali and T. Erdem, "Combining Haar Feature and skin color based classifiers for face detection," *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on,* 2011.

[8] K. Zuiderveld, "Contrast limited adaptive histogram equalization," *Academic Press Professional,* pp. 474-485 , 1994.

[9] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple," in *Accepted conference on computer vision and pattern recognition*, Cambridge, 2001.

[10] W. Tan, C. Chan, Y. Pratheepan and J. Condell, "A Fusion Approach for Efficient Human Skin Detection," *IEEE Transactions on Industrial Informatics,* vol. 8, pp. 138-147, 2012.

[11] Itseez, *Open Source Computer Vision Library,* https://github.com/itseez/opencv, 2015.

[12] S. Behnel and R. Bradshaw, "Cython: The Best of Both Worlds," *Computing in Science and Engineering,* vol. 13, no. 2, pp. 31-39, 2011.

[13] G. v. Rossum, *Python tutorial,* Amsterdam: Centrum voor Wiskunde en Informatica (CWI), 1995.

[14] A. A. S. O. Emir Buza, "Skin Detection Based on Image Color Segmentation with Histogram and K-Means Clustering," in *IEEE*, Bursa, Turkey, 2017.

[15] A. Kaur and B. Kranthi, "Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation," *International Journal of Applied Information Systems ,* vol. 3, no. 4, 2012.