2019 SAUPEC/RobMech/PRASA Conference
Bloemfontein, South Africa, January 28-30, 2019

A COMPARISON OF SKIN DETECTION ALGORITHMS FOR HAND GESTURE RECOGNITION

Timothy James McBride*, Nabeel Vandayar*, Kenneth John Nixon*

*School of Electrical & Information Engineering
University of the Witwatersrand
Johannesburg, South Africa
Email: Timothy.McBride@students.wits.ac.za, Ken.Nixon@wits.ac.za

***Abstract*—Hand gesture recognition software is becoming more accessible with the advances in depth cameras and sensors, but these sensors are still expensive and not freely available. A real time Hand Gesture Recognition software is designed to work with a low cost monocular web camera. Skin detection and skin extraction is a common form of image processing used for gesture recognition. A comparison of three different skin detection algorithms is performed. The three algorithms are: YCbCr thresholding, RGB-H-CrCb thresholding and KNN Classification. The results obtained for each algorithm show that the algorithms are unreliable with a low mean and a large standard deviation. It was concluded that the uncertainty of the accuracy of each algorithm reduces the effectiveness of the hand gesture recognition software and it is not implemented in the final design. Alternative skin detection algorithms are suggested to improve on the accuracies and latencies obtained.**

***Index Terms*—Human-Computer Interface, Image Processing, Image Segmentation, Skin Detection, Skin Thresholding**

I. INTRODUCTION

A Human-Computer Interface (HCI) is the way in which a human communicates with a computer and is commonly composed of specialised hardware such as a mouse and keyboard. This form of communication is unnatural and presents a barrier between the user and computer. Hand Gesture Recognition (HGR) software allows a user to communicate with a computer using hand gestures and is a more natural form of communication as it resembles human to human communication. A HCI that makes use of HGR presents various application scenarios. For example, playing games, corporate presentations and control of robotics in environments where typical forms of HCI are ineffective.

The primary challenge for HGR is that it normally requires specialised and expensive equipment e.g. wearable electromagnetic devices or depth sensing cameras. A form of HGR is implemented that makes use of a low cost monocular web camera. The system operates at real time and uses a Convolutional Neural Network (CNN) to classify various hand gestures performed by the user. This document describes and analyses the skin detection algorithms that were investigated to improve the accuracy of the HGR system developed.

Skin detection is the process of identifying the location of human skin within an image and is used to determine where hands and faces are located. These features can be extracted from the image to reduce the amount of random information that exists in the background of the image. This ideally improves the accuracy of the gesture classification output from the CNN, as dimensionality reduction has been performed. Three methods of skin extraction are tested: RGB-H-CrCb Thresholding, YCbCr Thresholding and K-Nearest Neighbour (KNN) Classification. The accuracy and latency of each method is tested and analysed using a ground truth dataset. These methods are chosen as they take into consideration different ethnicities and lighting conditions within the image.

II. BACKGROUND

A. Literature Review

A standard camera can be used for hand tracking and hand gesture recognition systems [1, 2]. The process that is outlined by Yeo and Zu provides a basic grounding for the implementation of a hand gesture recognition system [1, 2]. A similar process was used during the investigation of the HGR software. The results obtained by Yeo et al. show that a hand tracking system for mouse control is not a desirable feature as it is uncomfortable for the user [1]. These results were taken into consideration and a hand tracking system was not implemented.

The primary method for skin extraction is the use of skin colour thresholding, which is the process of extracting a particular region of colour that represents skin [3]. Skin colour thresholding methods that make use of different colour spaces can be used for skin extraction [3–6]. The most common of which is the YCbCr colour space due to the large correlation in the Cr and Cb channels for skin tones. A comparison of the YCbCr colour space to the alternative HSV colour space show that due to the slowly varying distribution of the HSV colour space that it is less effective at skin thresholding than the YCbCr space [3]. These results were taken into consideration and HSV thresholding is not used in the investigation. A

978-1-7281-0369-3/19/$31.00 ©2019 IEEE

thresholding technique presented by Kolkur et al. and bin Abdul Rahman et al. makes use of three different colour spaces, namely RGB, HSV and YCbCr [5, 6]. The results showed that the technique was worth investigating and the algorithm presented by bin Abdul Rahman et al. is implemented [6]. A measurement for the accuracy of skin extraction is determined after consideration of existing solutions [5].

A Gaussian Mixture Model (GMM) can be used for the prediction of skin pixels however it works in a similar way to the thresholding techniques and is more complicated to implement, it is determined to be out of scope for the investigation [7, 8]. An alternative dynamic approach is considered that makes use of the KNN Machine learning Algorithm. The KNN classifier has had success in determining melanoma skin lesions based off of image colour and texture [9]. The methodology is adapted for the use in skin pixel classification.

B. Colour Spaces

A Colour space is a mathematical representation of colour information. The space is divided into different channels. These channels contain numeric representations of colour components that can be combined together to show colour. Different colour spaces are used for different applications such as computer graphics, image processing and TV broadcasting [5]. For skin detection the colour spaces that were investigated are:

- Red, Green and Blue (RGB) Colour Space
- Hue, Saturation and Value (HSV) Colour Space
- Luminance and Chrominance (YCbCr) Colour Space

1) RGB Colour Space: RGB is the most common model used as it is normally the default colour space for storing digital images. The RGB colour space is used by computers, graphics and digital display systems [5]. This colour space is an additive colour model that has three components. These are the primary colours: Red (R), Green (G) and Blue (B). To form a specific colour the three channels are superimposed onto each other with some arbitrary intensity. This mixture of channels produces the desired colour [4]. Linear and non linear transformations are used to convert the RGB space into the other colour spaces available. *Fig. 1* shows the visual representation of this colour model, each channel is normally represented as integer values in a range from 0 to 255.

2) HSV Colour Space: The HSV colour model is more intuitive to how people view colour. The Hue (H) channel represents the specific type of colour that is desired. The Saturation (S) channel is then varied to change the corresponding colours defined by the Hue from unsaturated (grey) to fully saturated (no white component) [5]. The last channel, Value (V), represents the brightness of the image. The HSV colour model is represented as a cylinder shown in *Fig. 2*, this is because the H channel is in a range between 0° and 360° while the S and V channels are between 0 and 1.

3) YCbCr Colour Space: The YCbCr colour space uses a non-linear encoding of RGB to separate the luminance and chrominance components of the colour. This is commonly used in the digital video domain. Due to this representation of the colour space it is easy to remove redundant colour information from an image. This aspect of the colour model causes it to be used in image and video compression standards such as JPEG, MPEG1, MPEG2 and MPEG4 [5]. The model explicitly separates the luminance and chrominance into different channels. The luminance is stored in the Y channel while the chrominance is stored as two colour difference components Cb and Cr. Cb represents the difference between the B and Y while the Cr channel represents the difference between R and the Y. The colour model is shown in *Fig. 3.*

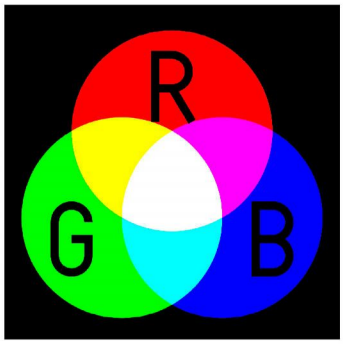

Fig. 1. The RGB colour space

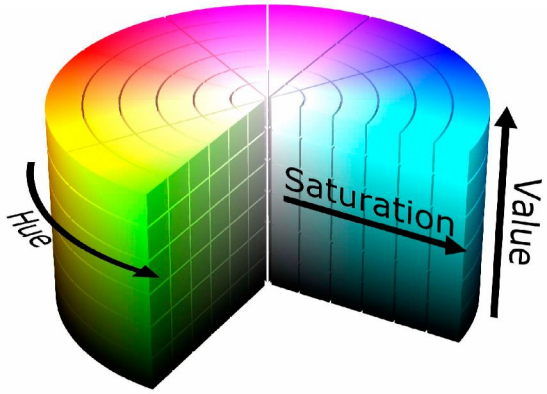

Fig. 2. The HSV colour space

III. SKIN DETECTION TECHNIQUES

A. YCbCr Skin Thresholding

Compared to other colour models YCbCr provides the best discrimination between skin and non skin pixels this is independent of ethnicity and luminance as the correlation is performed between the Cr and Cb channels only [3, 4]. The skin thresholding technique uses predefined threshold values to extract the region of colours that would usually correspond to skin. The thresholding values used in the implementation for the YCbCr thresholding are the values put forward by Basilio *et al* [4]. The YCbCr threshold range is shown in (1)

$$80 \leq Cb \leq 120 \cap 133 \leq Cr \leq 173 \quad (1)$$

B. RGB-H-CrCb Skin Thresholding

In the RGB space the colour of skin is not well defined, this is due to the luminance being contained in all three channels of the image. The histogram of the skin colours in the RGB space shows that there is a uniform spread across a large spectrum of values [6]. Using a combination of colour spaces for skin colour improves the classification of whether a pixel is skin or non skin. The thresholding technique proposed by bin Abdul Rahman *et al* [6] presents a combination of the RGB, HSV and YCbCr colour spaces. This form of thresholding improves on the discriminality between skin pixels and non-skin pixels[6]. The thresholding values presented by bin Abdul Rahman *et al* [6] are implemented. Thresholding rules are used to determine particular regions in each of the colour spaces used. These regions are combined together using logical AND operations.

C. KNN Skin Classification

K-Nearest Neighbour Classification is a simple supervised machine learning algorithm. The model works by storing the training data set and using the stored information to classify an input. The input is compared to the closest neighbour and is classified accordingly. The value of k determines the number of neighbours that the input is compared to. In a situation where k is greater than one a voting system is implemented and each neighbour assigns a label, the labels are counted and the input is classified as the most frequent label. The lower the value of k the more complex the model becomes [10]. A KNN classifier is used to classify each pixel of the input image as either skin or non skin. The model is trained using the skin segmentation dataset provided by Bhatt and Dhall [11]. The latency of the model is effected by the number of features within the model, since there are only three features being tested, the RGB values of the pixel, it is an appropriate technique for real time skin classification.

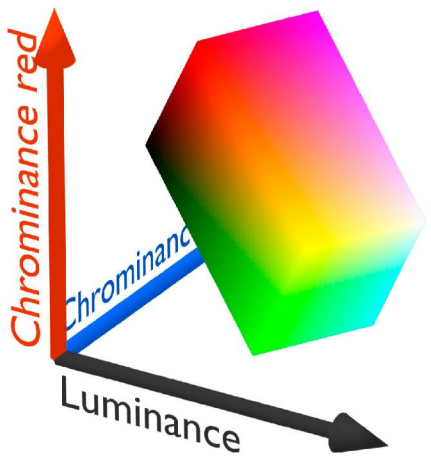

Fig. 3. The YCbCr colour space

IV. METHODOLOGY

Each of the skin detection algorithms were implemented using MATLAB and the accuracy and latency of the technique was tested using the *Pratheepan* dataset as ground truth [12]. The dataset contains a set of random images obtained from the internet used for human skin detection research. The dataset also contains ground truth for each of the images in the dataset. The images contain pictures of people with varying ethnicities and are captured with a wide range of cameras under different illuminations and is used as there is not an appropriate ground truth dataset specifically for hands. During the latency tests the images are resized to a 100×100 pixel image. This is done to remove the effect that image size has on the latency of the algorithms.

The resultant image mask obtained from the different algorithms is compared to the ground truth image. The number of True Positives (TP) and True Negatives (TN) are determined. TP are skin pixels that have been correctly classified as skin pixels while TN are non skin pixels that have been correctly classified as non skin pixels. The percentage accuracy of skin detected and non skin detected is determined using (2) and (3) respectively. The values determined are multiplied together to calculate the overall accuracy of the skin detection algorithm, as shown in (4). This calculation considers the skin detection and non skin detection equally and ensures that both need to have a high accuracy for the total accuracy of the algorithm to be high.

$$SDA = \frac{TP}{TSP} \tag{2}$$

Where:

SDA = Skin Detection Accuracy
TP = True Positives
TSP = Total Number of Skin Pixels

And:

$$NSDA = \frac{TN}{TNSP} \tag{3}$$

Where:

$NSDA$ = Non Skin Detection Accuracy
TN = True Negatives
$TNSP$ = Total Number of Non Skin Pixels

$$TotalAccuracy = SDA \times NSDA \times 100 \tag{4}$$

V. RESULTS

A. YCbCr Thresholding Accuracy

The YCbCr thresholding algorithm is effective at detecting skin within an image however it less effective at detecting non skin pixels correctly. The accuracy for the skin and non skin detection is shown in *Fig. 4*. The algorithm is unable to differentiate between skin and non skin for regions that are

very dark or very bright, this is likely caused by the exclusion of the luminance component in the predefined thresholding region. The total accuracy for YCbCr is shown in *Fig. 5*, it is clear from the results that the algorithm is inconsistent. This is largely due to the colour of the background and the lighting of the image. Images with backgrounds that resemble skin colour perform poorly while images with backgrounds that are non skin colours perform well.

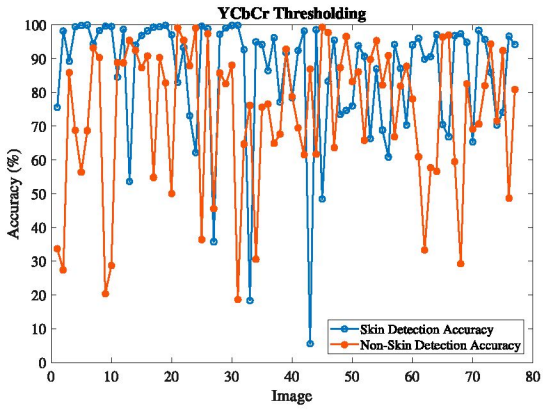

Fig. 4. The skin and non-skin detection accuracy for the YCbCr thresholding algorithm

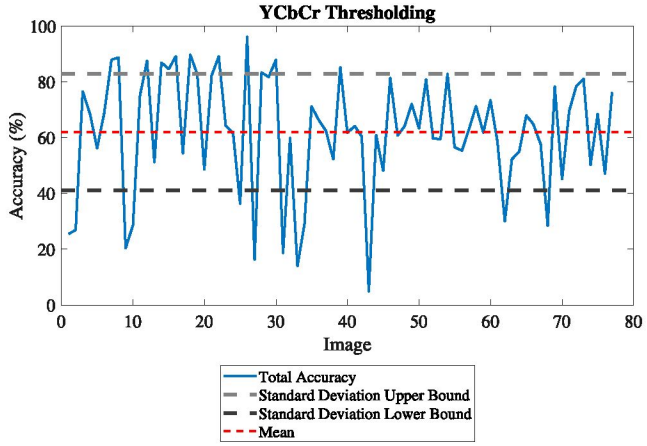

Fig. 5. The total accuracy for the YCbCr thresholding algorithm

B. RGB-H-CrCb Thresholding Accuracy

RGB-H-CrCb thresholding provides a combination of colour spaces in the skin detection algorithm. This algorithm performs effectively and is able to successfully extract skin regions from most of the images however the algorithm is primarily affected by the brightness of the image as it classifies white or bright areas as skin and dark areas as non-skin. This results in inaccuracies with images that have white backgrounds and with ethnicities that have a darker skin tone. The accuracy results obtained for the algorithm are shown in Fig. 6 and 7. The logical operation used in combining each thresholding region minimizes the amount of skin detected in the final output, this maximizes the number of non skin classifications for each thresholding region.

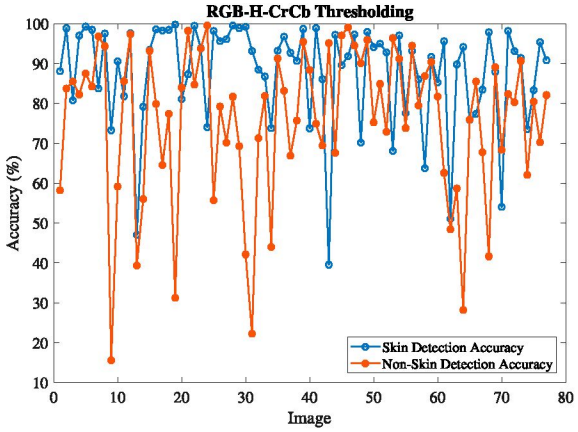

Fig. 6. The skin and non-skin detection accuracy for the RGB-H-CrCb thresholding algorithm

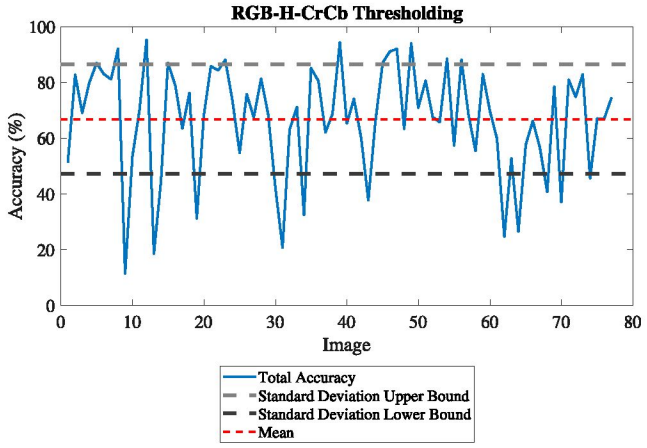

Fig. 7. The total accuracy for the RGB-H-CrCb thresholding algorithm

C. KNN Classification Accuracy

The KNN classification classifies pixels according to the training data provided, this limits the effectiveness of the algorithm. The model created is successfully able to classify skin and non-skin pixels based off of the RGB values of the image. The model is more effective at classifying non-skin pixels than skin pixels as shown in *Fig. 8*; this is a result of a bias within the training data as it contains more values for non-skin pixels than skin pixels. This bias will cause a negative impact on the output of the skin detector by causing skin pixels to be classified incorrectly as non skin pixels. The model has inaccuracies with images that have skin coloured backgrounds and is effected by the luminance of the image. These factors cause the model to perform poorly on specific images, For example, images that contain people with pale or light coloured skin. The overall accuracy for the KNN classifier is shown in *Fig. 9*.

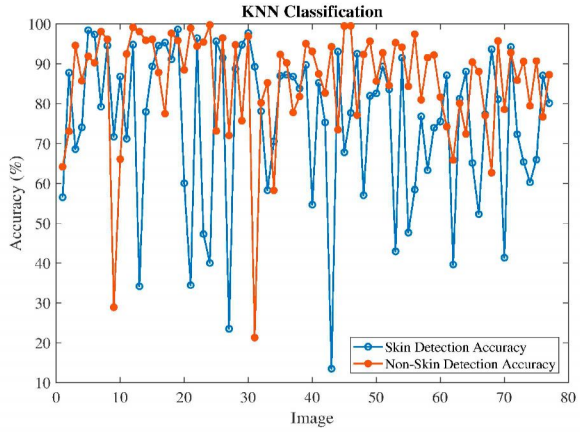

Fig. 8. The skin and non-skin detection accuracy for the KNN classification algorithm

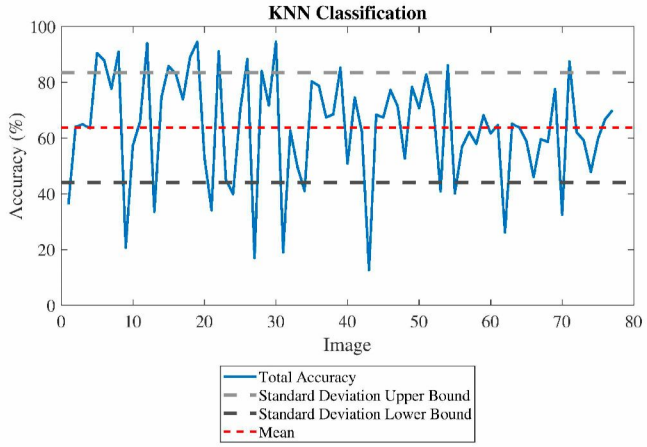

Fig. 9. The total accuracy for the KNN classification algorithm

D. Latency

The latency of the algorithms is a major factor in the application of HGR software. The system needs to operate in real time so the skin detection algorithm is required to have a small latency. The processing time for each of the algorithms analysed is measured and an average latency for the algorithm is determined. *Table I* shows the average latency for each algorithm while performing skin detection on the dataset of images. Each image is resized to a 100×100 pixel image, this is done to remove the effect that the image size has on the computational time of each algorithm.

TABLE I
AVERAGE LATENCY RESULTS FOR THE DIFFERENT SKIN DETECTION ALGORITHMS

Skin Detection Algorithm	Average Latency
RGB-H-CrCb Thresholding	0.0021 s
YCbCr Thresholding	0.0012 s
KNN Classification	0.0337 s

VI. DISCUSSION

The results show that each algorithm is able to detect skin within images to some success. The algorithms are inconsistent and have a low accuracy with large standard deviations. This is the result of varying skin tones due to the luminance of the image and the similarities of skin colour present in the backgrounds of the testing images. The algorithms tested all make use of colour thresholding or matching. This factor will limit the effectiveness of the algorithms. If the colours of the background resemble a shade of skin colour then the accuracy of the algorithm for that image will be reduced. This effect is more prominent in the thresholding algorithms as they have predefined threshold values. Morphological operations such as erosion and dilation can be applied to improve the accuracy of the algorithms. These operations are effective at removing small regions of false positives and false negatives but they are unable to remove large regions of false classifications. *Fig. 10* shows a comparison of the extracted images by each algorithm.

The YCrCb thresholding algorithm was less susceptible to the luminance component as it was extracted but it was still unable to distinguish between bright regions of skin and non skin. It also had the lowest latency of the different algorithms tested, this is an expected result as the YCbCr thresholding is the simplest of the algorithms implemented.

The RGB-H-CrCb thresholding had a similar problem as the YCrCb thresholding algorithm except that it considered all bright regions as skin. This resulted in white clothing or backgrounds to be left in the image. An advantage of this is that the darker regions in the background were successfully extracted from the image by the skin detector. This algorithm had the best mean accuracy and the lowest standard deviation but it is still inconsistent and unreliable.

The KNN Classifier is biased by the training data and would classify pixels as non skin more frequently, this caused the inconsistent results obtained. The classifier would be very accurate for some images but would perform poorly for others by classifying majority of the skin pixels as non skin. This can be improved by using a larger training dataset for the model or by using a different classifier all together.

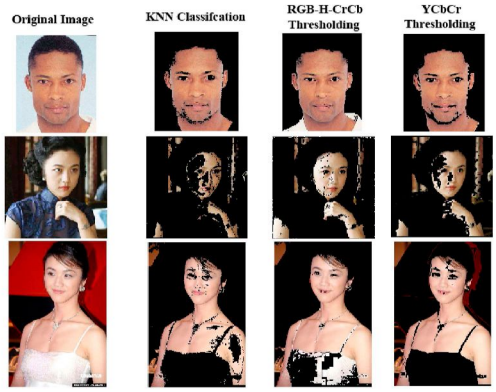

Fig. 10. A comparsion of the skin detection algorithms on particular images

VII. CONCLUSION

The RGB-H-CrCb thresholding algorithm had the highest average accuracy with the least standard deviation, however it is inconsistent and unable to classify skin reliably with varying illumination and backgrounds. The results obtained from each of the algorithms are very similar with minimal changes in accuracy and standard deviation. This shows that the predefined threshold values are not consistent for a general case model and an alternative solution is required. The skin detection algorithms reduce the accuracy of the hand gesture classification when implemented with the CNN. This is in comparison to the CNN using the direct RGB images captured by the web camera. A skin detection algorithm is not included in the final implementation due to these results.

FUTURE RECOMMENDATIONS

The algorithms are inaccurate and inconsistent with high standard deviations and low average accuracy. This is largely due to the predefined threshold values for skin detection an improved solution would be to investigate into an adaptive thresholding technique that is able vary the threshold values based on the viewed image. Mittal et al. make use of a face detector to identify a range of colours of the skin in an image [13]. This removes the effect of unusual lighting in the image as the range of skin colours can be used to determine optimal threshold values specific to that image. Alternatively different machine learning algorithms can be implemented with different colour spaces as additional inputs, by doing this the algorithm should be less susceptible to the flaws of a particular colour space. Probabilistic models have also been shown to have improved success and can be implemented. This includes Gaussian mixture models and Bayesian classifiers. Lastly a ground truth dataset that is primarily composed of hands should be developed to obtain a more acurate representation of the algorithms for the purpose of hand gesture recognition.

ACKNOWLEDGEMENT

The author would like to thank Dr Klein, from the School of Computer Science and Applied Mathematics, for their guidance with the project as well as Opti-Num Solutions for their support and for providing access to MATLAB and the various toolboxes used in the investigation.

REFERENCES

[1] H.-S. Yeo, B.-G. Lee, and H. Lim. "Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware." *Multimedia Tools and Applications*, vol. 74, no. 8, pp. 2687–2715, 2015.

[2] P. Xu. "A Real-time Hand Gesture Recognition and Human-Computer Interaction System." *arXiv preprint arXiv:1704.07296*, 2017.

[3] D. Marius, S. Pennathur, and K. Rose. "Face detection using color thresholding and eigenimage template matching." *Digital Image Processing project, Stanford University*, 2003.

[4] J. A. M. Basilio, G. A. Torres, G. S. Pérez, L. K. T. Medina, and H. M. P. Meana. "Explicit image detection using YCbCr space color model as skin detection." *Applications of Mathematics and Computer Engineering*, pp. 123–128, 2011.

[5] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia. "Human skin detection using RGB, HSV and YCbCr color models." *arXiv preprint arXiv:1708.02694*, 2017.

[6] N. A. bin Abdul Rahman, K. C. Wei, and J. See. "Rgb-h-cbcr skin colour model for human face detection." *Faculty of Information Technology, Multimedia University*, vol. 4, 2007.

[7] W. R. Tan, C. S. Chan, P. Yogarajah, and J. Condell. "A fusion approach for efficient human skin detection." *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 138–147, 2012.

[8] Q. Huynh-Thu, M. Meguro, and M. Kaneko. "Skin-color extraction in images with complex background and varying illumination." In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pp. 280–285. IEEE, 2002.

[9] L. Ballerini, R. B. Fisher, B. Aldridge, and J. Rees. "A color and texture based hierarchical K-NN approach to the classification of non-melanoma skin lesions." In *Color Medical Image Analysis*, pp. 63–86. Springer, 2013.

[10] A. C. Müller, S. Guido, et al. *Introduction to machine learning with Python: a guide for data scientists.* " O'Reilly Media, Inc.", 2016.

[11] A. D. Rajen Bhatt. "'Skin Segmentation Dataset'." UCI Machine Learning Repository.

[12] A. Albiol, L. Torres, and E. J. Delp. "Optimum color spaces for skin detection." In *ICIP (1)*, pp. 122–124. 2001.

[13] A. Mittal, A. Zisserman, and P. H. Torr. "Hand detection using multiple proposals." In *BMVC*, pp. 1–11. Citeseer, 2011.